
SunFounder Elite Explorer Kit

www.sunfounder.com

07.04.2024

Inhaltsverzeichnis

1	Erfahren Sie mehr über die Komponenten in Ihrem Kit	3
1.1	Komponentenliste	3
1.2	Einführung in die Komponenten	3
2	Einstieg in Arduino	65
2.1	Was ist Arduino?	65
2.2	Was kann Arduino tun?	66
2.3	Wie man ein Arduino-Projekt aufbaut	66
3	Code herunterladen	101
4	Grundlegende Projekte	103
4.1	Fotowiderstand	103
4.2	Thermistor	106
4.3	Bodenfeuchtigkeit	109
4.4	Kippschalter	111
4.5	PIR-Bewegungssensormodul	113
4.6	Ultraschall	116
4.7	Feuchtigkeitssensor-Modul	119
4.8	RFID-RC522 Modul	123
4.9	GY-87 IMU Modul	127
4.10	LED-Modul	140
4.11	RGB-LED	142
4.12	WS2812 RGB-LED-Streifen	148
4.13	7-Segment-Anzeige	151
4.14	I2C LCD1602	155
4.15	OLED	158
4.16	Aktiver Summer	161
4.17	Passiver Summer	165
4.18	Audio-Modul und Lautsprecher	169
4.19	Taster	173
4.20	Potentiometer	176
4.21	Joystick-Modul	180
4.22	Tastenfeld	182
4.23	Infrarot-Empfänger	186
4.24	MPR121	191
4.25	Motor	196

4.26	Wasserpumpe	199
4.27	Schrittmotor	201
4.28	Servo	206
4.29	Relais	209
4.30	74HC595	212
5	Erkunden Sie Arduino® UNO R4 WiFi	217
5.1	Wi-Fi	217
5.2	Bluetooth	223
5.3	Echtzeituhr	231
5.4	12x8 LED-Matrix	233
5.5	Digital-Analog-Wandler (DAC)	239
5.6	USB HID	242
6	Spaßprojekte	247
6.1	Willkommen	247
6.2	Obstklavier	250
6.3	HueDial	253
6.4	Lichtempfindliches Array	256
6.5	Digitaler Würfel	259
6.6	Smarter Ventilator	262
6.7	Smarter Mülleimer	265
6.8	Pflanzenmonitor	267
6.9	Zugangskontrollsystem	270
6.10	SPIEL - Zahl Erraten	273
6.11	SPIEL - Flucht	276
6.12	SPIEL - Pong	278
6.13	SPIEL - Snake	281
7	IoT-Projekte	285
7.1	Einfacher Webserver	285
7.2	Arduino IoT Cloud	287
7.3	Sicherheitssystem über IFTTT	307
7.4	Cloud-Anrufsystem mit MQTT	318
7.5	CherryLight	323
7.6	WeatherTime Bildschirm	327
7.7	Bluetooth-Nachrichtenbox	333
7.8	Bluetooth-Umweltmonitor	340
8	Anhang	353
8.1	Aktualisieren der Radio-Modul-Firmware auf Ihrem UNO R4 WiFi-Board	353
8.2	Wie scannt und erkennt man I2C-Adressen?	355
9	Vielen Dank!	359

Mit SunFounder wird das Programmieren von Arduino aufgewertet. Das Elite Explorer Kit, kombiniert mit dem bahnbrechenden Uno R4 WiFi, ist die ultimative All-in-One-Lösung, um Arduino zu meistern und fortgeschrittene DIY-Elektronikprojekte zu bauen. Bestellen Sie noch heute und lassen Sie Ihre Kreativität in die Höhe schnellen!

Inhaltsverzeichnis

Erfahren Sie mehr über die Komponenten in Ihrem Kit

1.1 Komponentenliste

Bitte überprüfen Sie nach dem Öffnen des Pakets, ob die Anzahl der Komponenten mit der Produktbeschreibung übereinstimmt und ob alle Komponenten in einwandfreiem Zustand sind.

- Komponentenliste

1.2 Einführung in die Komponenten

Im Folgenden finden Sie eine Einführung zu jeder Komponente, die das Funktionsprinzip der Komponente und die entsprechenden Projekte enthält.

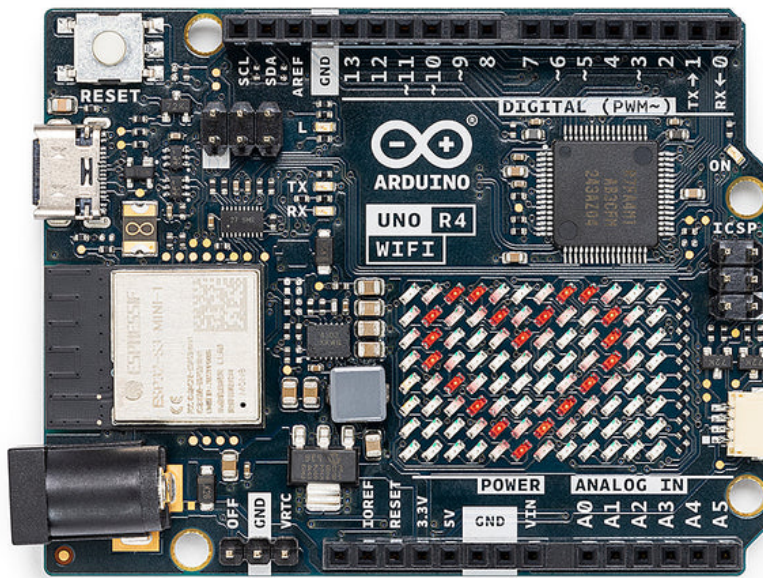
Grundlegendes

1.2.1 Arduino Uno R4 WiFi

Übersicht

Das Arduino UNO R4 WiFi repräsentiert den Höhepunkt von IoT und drahtloser Innovation. Ausgestattet mit der Leistung des RA4M1-Mikrocontrollers von Renesas und weiter verstärkt durch einen ESP32-S3-Koprozessor, wurde dieses Board akribisch entwickelt, um den sich entwickelnden Anforderungen moderner Maker gerecht zu werden. Egal, ob Sie neu in der Arduino-Welt sind oder ein erfahrener Technikenthusiast, das UNO R4 WiFi garantiert erstklassige Leistung und behält dabei das vertraute Formfaktor und die 5 V Betriebsspannung bei.

Das UNO R4 WiFi tritt als Symbol für Konnektivität, Effizienz und Einfallsreichtum weiter in den Arduino-Bereich ein.



Hier ist, was das Arduino UNO R4 WiFi bietet:

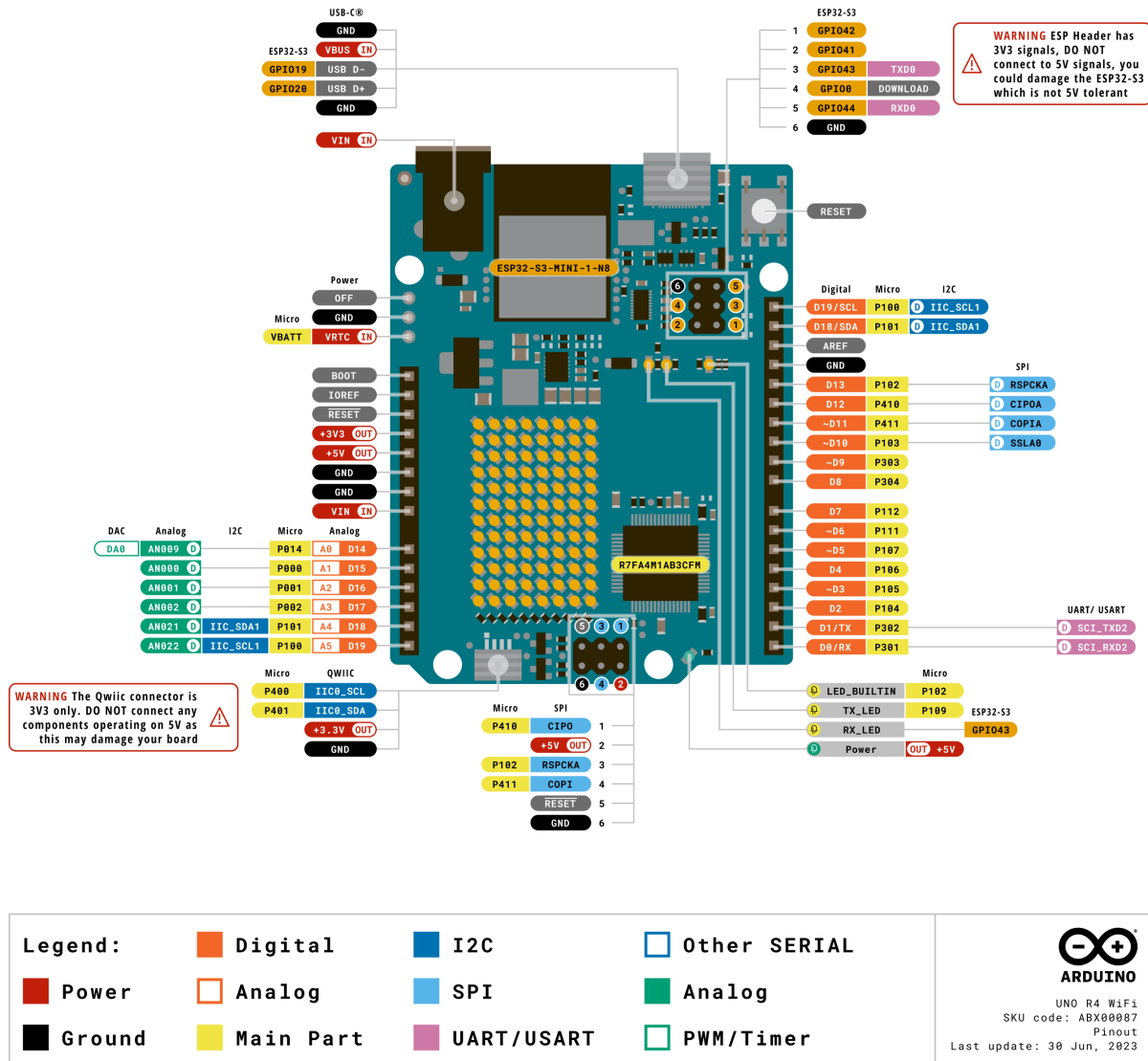
- **Nahtlose Integration in das UNO-Ökosystem:** Treu seinem Erbe, garantiert das UNO R4 WiFi Kompatibilität mit dem ikonischen UNO-Formfaktor, Pinout und der 5 V Betriebsspannung. Der Übergang von früheren Versionen ist mühelos, dank des harmonischen Designs und des umfangreichen Arduino-UNO-Ökosystems.
- **Leistungsfähiger Speicher und Verarbeitung:** Treten Sie in eine Welt schnellerer Berechnungen und komplexer Projekte ein. Das UNO R4 WiFi verfügt nicht nur über erweiterten Speicher, sondern arbeitet auch mit einer dreimal schnelleren Taktfrequenz, um sicherzustellen, dass Ihre Projekte reibungslos und effizient laufen.
- **Vielfältige On-Board-Peripheriegeräte:** Vom 12-Bit-DAC und CAN-Bus bis hin zu einem OP-Verstärker und einem einzigartigen SWD-Port ist das UNO R4 WiFi mit Funktionen ausgestattet, die Ihre Projekt-Fähigkeiten erhöhen. Tauchen Sie ein in eine Welt unendlicher Möglichkeiten und entfesseln Sie Ihre Kreativität.
- **Konnektivität vom Feinsten:** Mit integriertem Wi-Fi® und Bluetooth® Low Energy ebnet das UNO R4 WiFi den Weg zum Internet der Dinge. Ob Sie ein Smart-Home-System oder ein interaktives Dashboard erstellen, dieses Board unterstützt Sie dabei.
- **Interaktive 12×8 LED-Matrix:** Beleuchten Sie Ihre Projekte mit dynamischen Animationen oder Echtzeit-Sensordatenvisualisierung, ohne externe Hardware zu benötigen.
- **Fortgeschrittene Sicherheitsmechanismen:** Die Fähigkeit des Boards, potenziell schädliche Operationen wie Division durch Null zu erkennen und zu verhindern, sorgt für ein nahtloses Erlebnis. Zusätzlich erhalten Sie durch detailliertes Feedback auf dem seriellen Monitor stets Einblick in den Ablauf.
- **Qwiic-Anschluss für schnelles Prototyping:** Erweitern Sie Ihren Projektumfang mit dem Qwiic-Anschluss. Mit einer Vielzahl von I2C-kompatiblen Modulen wird das Prototyping zum Kinderspiel.

Treten Sie mit dem Arduino UNO R4 WiFi in die Zukunft des Machens ein. Ob Sie drahtlose Funktionalitäten integrieren, die weite Landschaft des IoT erkunden oder einfach Ihr bestehendes Setup aufrüsten möchten, dieses Board ist der ideale Partner für Ihre bevorstehenden Unternehmungen.

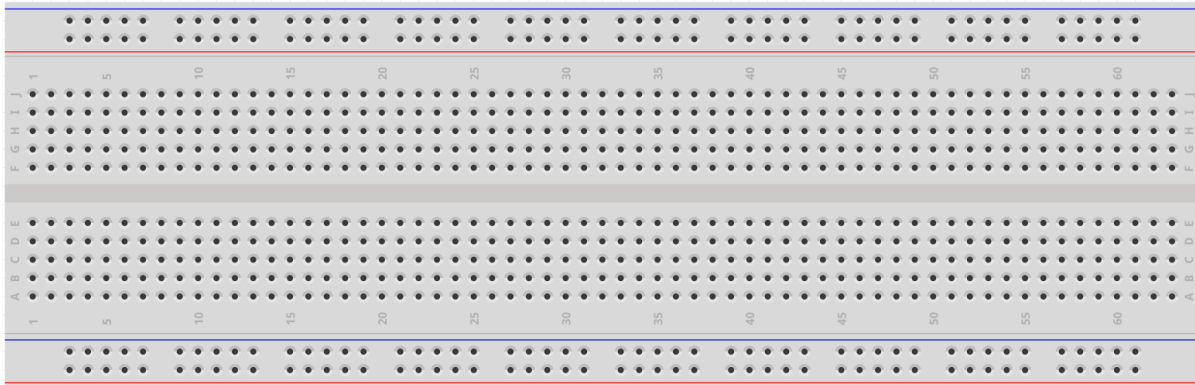
Technische Spezifikationen

Board	Name	Arduino® UNO R4 WiFi
Mikrocontroller	Renesas RA4M1 (Arm® Cortex®-M4)	
USB	USB-C®	Programmieranschluss
Pins	Digitale I/O-Pins	14
Pins	Analoge Eingangspins	6
	DAC	1
	PWM-Pins	6
Kommunikation	UART	Ja, 1x
	I2C	Ja, 1x
	SPI	Ja, 1x
	CAN	Ja 1 CAN-Bus
Energie	Betriebsspannung des Schaltkreises	5 V (ESP32-S3 ist 3.3 V)
	Eingangsspannung (VIN)	6-24 V
	DC-Strom pro I/O-Pin	8 mA
Taktgeschwindigkeit	Hauptkern v	48 MHz
	ESP32-S3	bis zu 240 MHz
Speicher	RA4M1	256 kB Flash, 32 kB RAM
	ESP32-S3	384 kB ROM, 512 kB SRAM
Abmessungen	Breite	68.85 mm
	Länge	53.34 mm

Pinout



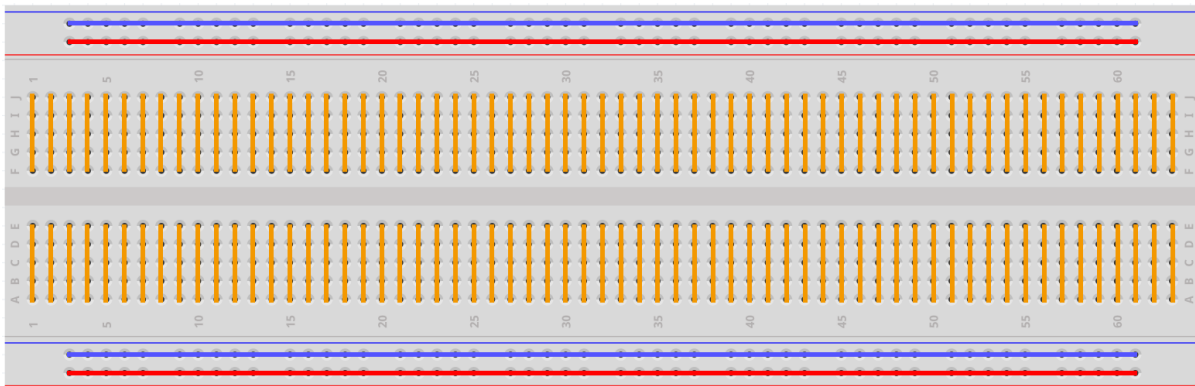
1.2.2 Steckbrett



Ein Steckbrett ist eine Basis für den Prototypenbau in der Elektronik. Ursprünglich bezog sich das Wort auf ein buchstäbliches Brotbrett, ein poliertes Holzstück, das zum Brotschneiden verwendet wurde. In den 1970er Jahren wurde das lötfreie Steckbrett (auch als Steckplatine oder Terminal-Array-Board bekannt) eingeführt, und heute wird der Begriff „Steckbrett“ allgemein für diese verwendet.

Es wird verwendet, um Schaltkreise schnell aufzubauen und zu testen, bevor ein Schaltungsdesign abgeschlossen wird. Es hat viele Löcher, in die die oben genannten Komponenten wie ICs und Widerstände sowie Jumperkabel eingefügt werden können. Das Steckbrett ermöglicht es Ihnen, Komponenten leicht einzustecken und zu entfernen.

Das Bild zeigt die innere Struktur eines Steckbretts. Obwohl diese Löcher auf dem Steckbrett unabhängig voneinander zu sein scheinen, sind sie intern tatsächlich über Metallstreifen miteinander verbunden.



Wenn Sie mehr über Steckbretter erfahren möchten, schauen Sie hier:

1.2.3 Widerstand



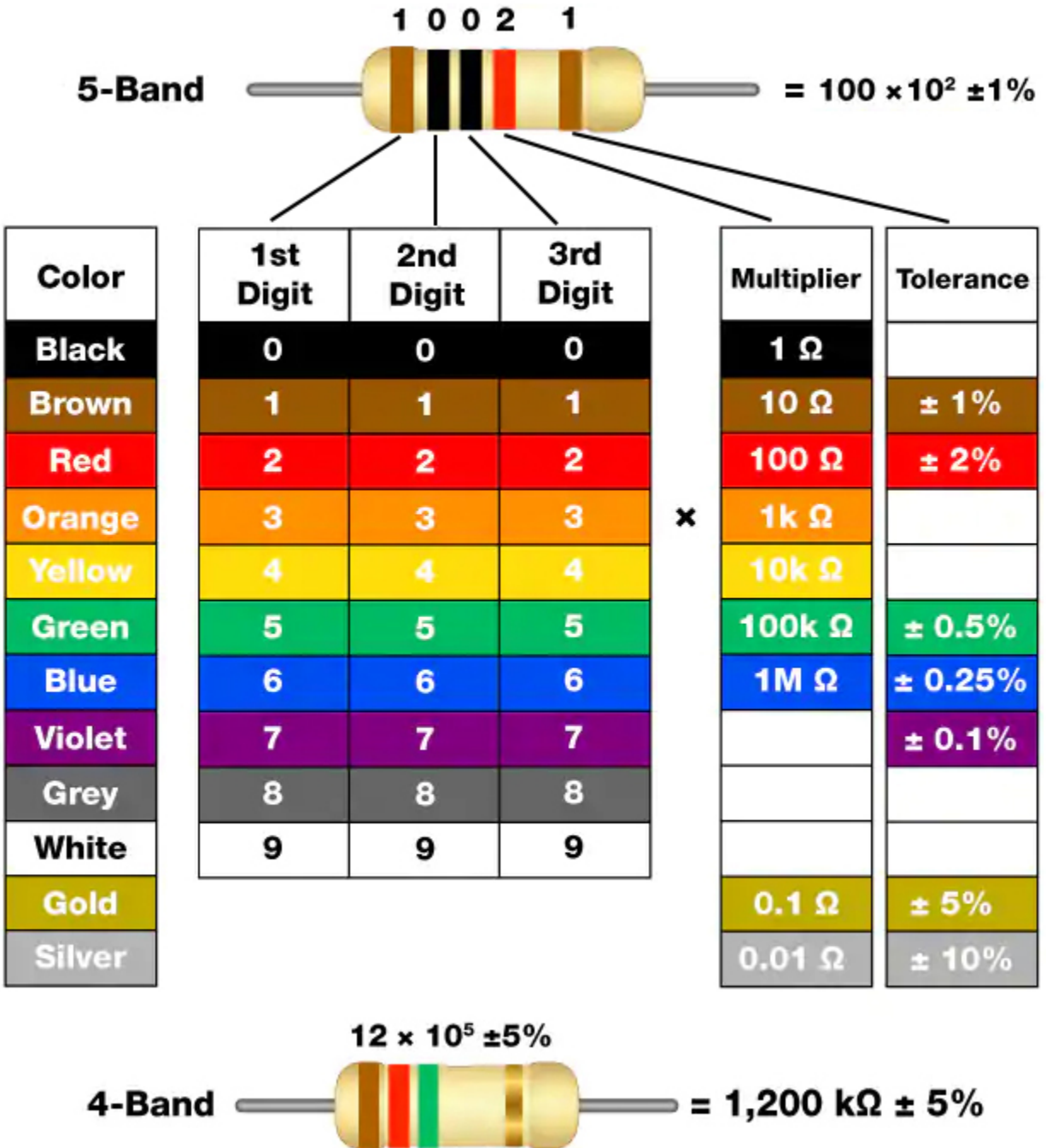
Ein Widerstand ist ein elektronisches Bauelement, das den Stromfluss in einem Zweig begrenzen kann. Ein Festwiderstand ist eine Art Widerstand, dessen Widerstandswert nicht verändert werden kann, während der Widerstandswert eines Potentiometers oder eines variablen Widerstands angepasst werden kann.

Es gibt zwei allgemein verwendete Schaltungssymbole für Widerstände. Normalerweise ist der Widerstandswert darauf markiert. Wenn Sie also diese Symbole in einer Schaltung sehen, steht dies für einen Widerstand.



ist die Einheit des Widerstands, und größere Einheiten umfassen K, M usw. Ihre Beziehung kann wie folgt dargestellt werden: $1\text{ M} = 1000\text{ K}$, $1\text{ K} = 1000$. Normalerweise ist der Wert des Widerstands darauf markiert.

Bei der Verwendung eines Widerstands müssen wir zuerst seinen Widerstandswert kennen. Hier sind zwei Methoden: Sie können die farbigen Bänder auf dem Widerstand beobachten oder ein Multimeter verwenden, um den Widerstand zu messen. Es wird empfohlen, die erste Methode zu verwenden, da sie bequemer und schneller ist.



Wie auf der Karte gezeigt, steht jede Farbe für eine Zahl.

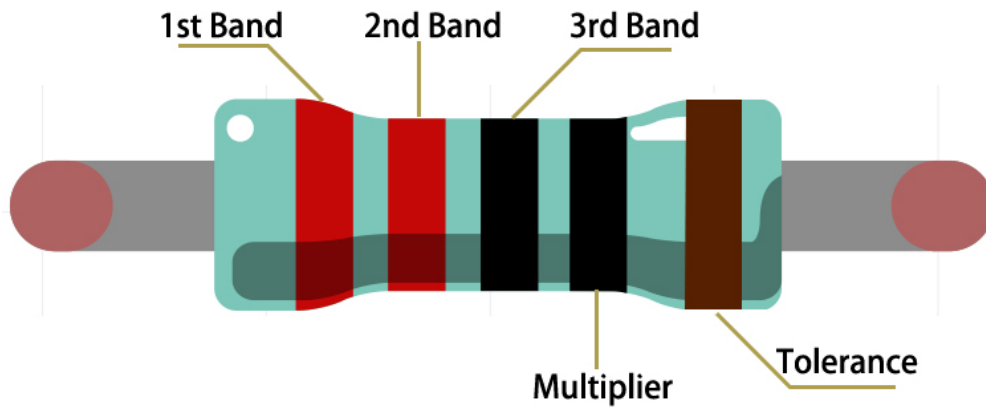
Schwarz	Braun	Rot	Orange	Gelb	Grün	Blau	Violett	Grau	Weiß	Gold	Silber
0	1	2	3	4	5	6	7	8	9	0,1	0,01

Häufig verwendet werden 4- und 5-Band-Widerstände, bei denen es 4 bzw. 5 farbige Bänder gibt.

Normalerweise, wenn Sie einen Widerstand erhalten, kann es schwierig sein zu entscheiden, an welchem Ende Sie mit dem Lesen der Farbe beginnen sollen. Der Tipp ist, dass der Abstand zwischen dem 4. und 5. Band vergleichsweise größer sein wird.

Daher können Sie den Abstand zwischen den beiden farbigen Bändern an einem Ende des Widerstands beobachten; wenn dieser größer ist als jeder andere Bandabstand, dann können Sie von der gegenüberliegenden Seite lesen.

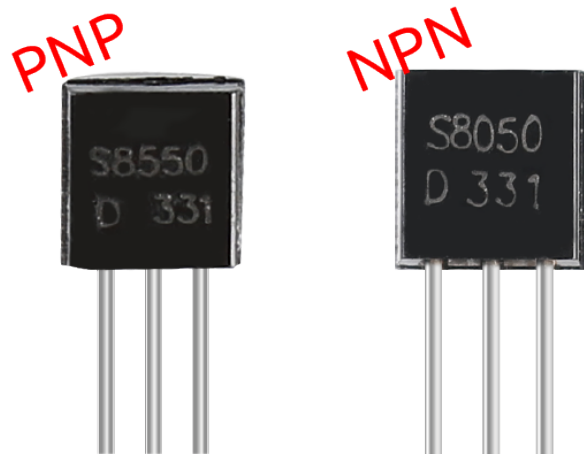
Lassen Sie uns sehen, wie man den Widerstandswert eines 5-Band-Widerstands wie unten gezeigt liest.



Also sollte bei diesem Widerstand der Widerstandswert von links nach rechts gelesen werden. Der Wert sollte in diesem Format sein: 1. Band 2. Band 3. Band $\times 10^{\text{Multiplikator}}$ () und der zulässige Fehler ist $\pm \text{Toleranz}\%$. So ist der Widerstandswert dieses Widerstands 2 (rot) 2 (rot) 0 (schwarz) $\times 10^0$ (schwarz) = 220 , und der zulässige Fehler ist $\pm 1\%$ (braun).

Sie können mehr über Widerstände auf Wiki erfahren: [Widerstand – Wikipedia](#).

1.2.4 Transistor



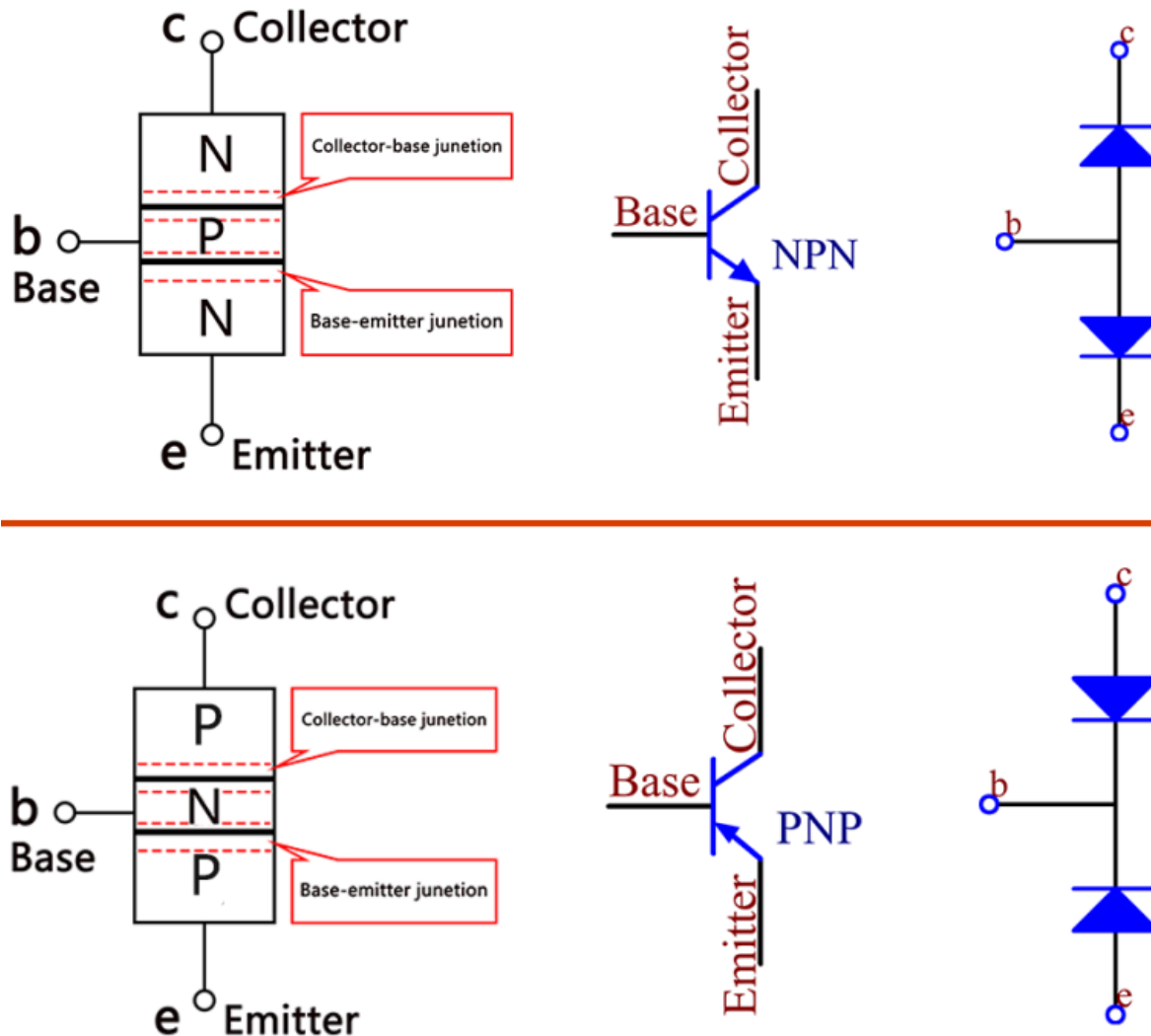
Ein Transistor ist ein Halbleitergerät, das den Strom durch Strom steuert. Seine Funktion besteht darin, schwache Signale in Signale mit größerer Amplitude zu verstärken und wird auch als kontaktloser Schalter verwendet.

Ein Transistor ist eine dreischichtige Struktur, die aus P-Typ- und N-Typ-Halbleitern besteht. Diese bilden intern drei Regionen. Die dünnere in der Mitte ist die Basisregion; die anderen beiden sind entweder N-Typ oder P-Typ – die kleinere Region mit intensiven Majoritätsträgern ist die Emitterregion, während die andere die Kollektorregion ist. Diese Zusammensetzung ermöglicht es dem Transistor, als Verstärker zu fungieren. Aus diesen drei Regionen entstehen jeweils drei Pole, die Basis (b), Emitter (e) und Kollektor (c) genannt werden. Sie bilden zwei P-N-Übergänge, nämlich den Emitterübergang und den Kollektorübergang. Die Richtung des Pfeils im Transistorschaltungssymbol gibt die des Emitterübergangs an.

- [P-N-Übergang – Wikipedia](#)

Basierend auf dem Halbleitertyp können Transistoren in zwei Gruppen eingeteilt werden, die NPN- und PNP-Typen. Aus der Abkürzung können wir entnehmen, dass die ersteren aus zwei N-Typ-Halbleitern und einem P-Typ bestehen und dass die letzteren das Gegenteil sind. Siehe die Abbildung unten.

Bemerkung: Der s8550 ist ein PNP-Transistor und der s8050 ist der NPN-Typ. Sie sehen sehr ähnlich aus, und wir müssen sorgfältig prüfen, um ihre Etiketten zu sehen.



Wenn ein High-Level-Signal durch einen NPN-Transistor fließt, wird er aktiviert. Ein PNP-Typ benötigt jedoch ein Low-Level-Signal zur Steuerung. Beide Transistortypen werden häufig für kontaktlose Schalter verwendet, wie in diesem Experiment.

Stellen Sie den Transistor mit der Etikettenseite zu uns und den Pins nach unten. Die Pins von links nach rechts sind Emitter(e), Basis(b) und Kollektor(c).



-
-

Beispiel

- *Relais* (Grundprojekt)
- *Aktiver Summer* (Grundprojekt)
- *Passiver Summer* (Grundprojekt)

1.2.5 Kondensator





Ein Kondensator, bezieht sich auf die Menge der Ladungsspeicherung unter einer gegebenen Potentialdifferenz, bezeichnet als C, und die internationale Einheit ist Farad (F). Allgemein gesagt, bewegen sich elektrische Ladungen unter Kraft in einem elektrischen Feld. Wenn sich ein Medium zwischen Leitern befindet, wird die Bewegung der elektrischen Ladungen behindert und die elektrischen Ladungen sammeln sich auf den Leitern an, was zur Ansammlung von elektrischen Ladungen führt.

Die Menge der gespeicherten elektrischen Ladungen wird als Kapazität bezeichnet. Da Kondensatoren eine der am häufigsten verwendeten elektronischen Komponenten in elektronischen Geräten sind, werden sie weit verbreitet in Gleichstromisolierung, Kopplung, Bypass, Filterung, Abstimmkreisen, Energieumwandlung und Steuerungsschaltungen eingesetzt. Kondensatoren werden in Elektrolytkondensatoren, Festkörperkondensatoren usw. unterteilt.

Nach Materialeigenschaften können Kondensatoren unterteilt werden in: Aluminiumelektrolytkondensatoren, Folienkondensatoren, Tantal-Kondensatoren, Keramikkondensatoren, Superkondensatoren usw.

In diesem Kit werden Keramik- und Elektrolytkondensatoren verwendet.

- [Keramikkondensator – Wikipedia](#)
- [Elektrolytkondensator – Wikipedia](#)

Auf den Keramikkondensatoren befinden sich 103- oder 104-Etiketten, die den Kapazitätswert darstellen, $103=10 \times 10^3 \text{pF}$, $104=10 \times 10^4 \text{pF}$

Einheitenumrechnung

$$1\text{F}=10^3\text{mF}=10^6\text{uF}=10^9\text{nF}=10^{12}\text{pF}$$

Beispiel

- [Taster](#) (Grundprojekt)

1.2.6 Diode

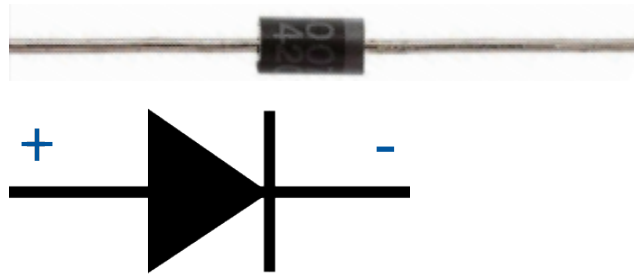
Eine Diode ist eine elektronische Komponente mit zwei Elektroden. Sie ermöglicht es dem Strom, nur in eine Richtung zu fließen, was oft als „Gleichrichter“-Funktion bezeichnet wird. Somit kann eine Diode als elektronische Version eines Rückschlagventils betrachtet werden.

Aufgrund ihrer unidirektionalen Leitfähigkeit wird die Diode in fast allen elektronischen Schaltungen einer gewissen Komplexität verwendet. Sie ist eines der ersten Halbleiterbauelemente und hat ein breites Anwendungsspektrum.

Je nach Einsatzklassifizierung kann sie in Detektor-Dioden, Gleichrichter-Dioden, Begrenzer-Dioden, Spannungsregler-Dioden usw. unterteilt werden.

Gleichrichter-Dioden und Spannungsregler-Dioden sind in diesem Kit enthalten.

Gleichrichter-Diode



Eine Gleichrichter-Diode ist eine Halbleiterdiode, die verwendet wird, um Wechselstrom (AC) in Gleichstrom (DC) unter Verwendung der Gleichrichterbrücken-Anwendung zu gleichrichten. Die Alternative der Gleichrichter-Diode durch die Schottky-Barriere wird hauptsächlich in der Digitaltechnik geschätzt. Diese Diode ist in der Lage, Stromwerte zu leiten, die von mA bis zu einigen kA reichen und Spannungen bis zu einigen kV.

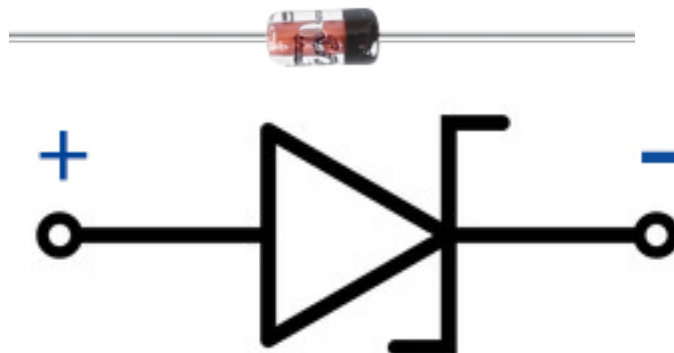
Die Gestaltung von Gleichrichter-Dioden kann mit Siliziummaterial erfolgen und sie sind in der Lage, hohe elektrische Stromwerte zu leiten. Diese Dioden sind nicht berühmt, aber es werden noch Ge- oder Galliumarsenid-basierte Halbleiterdioden verwendet. Ge-Dioden haben eine geringere zulässige Rückwärtsspannung sowie eine geringere zulässige Sperrschichttemperatur. Die Ge-Diode hat gegenüber der Si-Diode den Vorteil eines niedrigen Schwellenspannungswerts bei Betrieb in Vorwärtsrichtung.

•

Zener-Diode

Eine Zener-Diode ist eine spezielle Art von Diode, die zuverlässig den Stromfluss „rückwärts“ ermöglicht, wenn eine bestimmte festgelegte Rückwärtsspannung, bekannt als Zener-Spannung, erreicht wird.

Diese Diode ist ein Halbleiterbauelement, das einen sehr hohen Widerstand bis zur kritischen Rückwärtsdurchbruchspannung aufweist. An diesem kritischen Durchbruchpunkt verringert sich der Rückwärts-Widerstand auf einen sehr kleinen Wert, und der Strom steigt, während die Spannung in diesem Bereich mit niedrigem Widerstand konstant bleibt.



-

Beispiel

- *Relais* (Grundprojekt)

1.2.7 Jumperkabel

Kabel, die zwei Anschlüsse verbinden, werden als Jumperkabel bezeichnet. Es gibt verschiedene Arten von Jumperkabeln. Hier konzentrieren wir uns auf diejenigen, die auf Steckbrettern verwendet werden. Unter anderem werden sie verwendet, um elektrische Signale von irgendeinem Punkt auf dem Steckbrett zu den Ein-/Ausgangspins eines Mikrocontrollers zu übertragen.

Jumperkabel werden eingesteckt, indem ihre „Endanschlüsse“ in die vorgesehenen Steckplätze auf dem Steckbrett eingeführt werden, unter dessen Oberfläche sich einige Reihen paralleler Platten befinden, die die Steckplätze in Gruppen von Reihen oder Spalten verbinden, je nach Bereich. Die „Endanschlüsse“ werden ohne Löten in die bestimmten Steckplätze eingeführt, die in dem spezifischen Prototyp verbunden werden müssen.

Es gibt drei Arten von Jumperkabeln: Weiblich-zu-Weiblich, Männlich-zu-Männlich und Männlich-zu-Weiblich. Der Grund, warum wir es Männlich-zu-Weiblich nennen, liegt daran, dass es an einem Ende eine hervorstehende Spitze sowie ein versenktes weibliches Ende hat. Männlich-zu-Männlich bedeutet, dass beide Seiten männlich sind, und Weiblich-zu-Weiblich bedeutet, dass beide Enden weiblich sind.

Male-to-Female



Male-to-Male



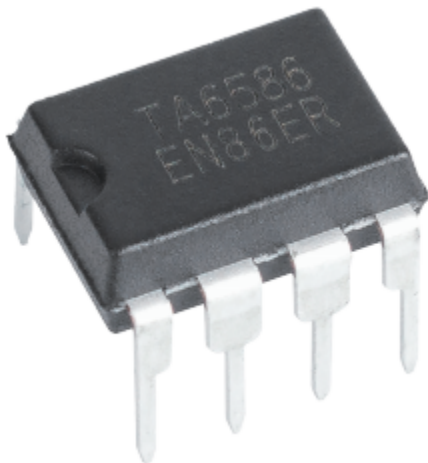
Female-to-Female



Mehr als ein Typ von ihnen kann in einem Projekt verwendet werden. Die Farbe der Jumperkabel ist unterschiedlich, aber das bedeutet nicht, dass ihre Funktion entsprechend unterschiedlich ist; es ist nur so gestaltet, um die Verbindung zwischen jedem Schaltkreis besser zu identifizieren.

Chip

1.2.8 TA6586 - Motorsteuerungs-Chip

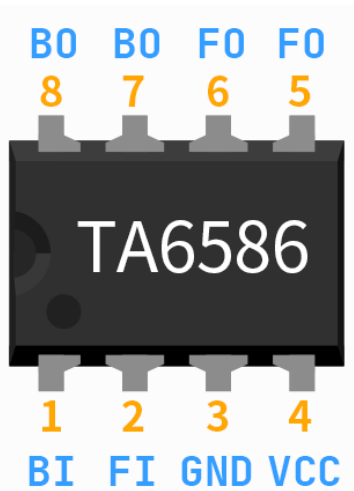


TA6586 ist ein monolithischer IC, der für das Ansteuern von bidirektionalen Gleichstrommotoren entwickelt wurde. Er verfügt über zwei Logikeingangspins zur Steuerung der Richtung, vorwärts und rückwärts. Die Schaltung weist eine gute Störfestigkeit, einen geringen Standby-Strom und einen niedrigen Ausgangssättigungsspannungsabfall auf. Sie hat eine eingebaute Klemmdiode, um den Aufprall der Freisetzung des induktiven Laststroms umzukehren, was sie in der Anwendung beim Antrieb von Relais, Gleichstrommotoren, Schrittmotoren oder beim Schalten von Leistung sicher und zuverlässig macht. TA6586 eignet sich für Spielzeugfahrzeuge, ferngesteuerte Flugzeugmotoren, automatische Ventilmotoren, elektromagnetische Schlossantriebe, Präzisionsinstrumente und andere Schaltungen.

Merkmale

- Geringer Standby-Strom: 2uA
- Breiter Versorgungsspannungsbereich
- Integrierte Bremsfunktion
- Thermischer Abschaltenschutz
- Überstrombegrenzung und Kurzschlusschutzfunktion
- DIP8 Pb-freies Gehäuse.

Pin-Funktion



Pin NO	Name	Function
1	BI	Backward input
2	FI	Forward input
3	GND	Ground
4	Vcc	Vcc
5, 6	FO	Forward output
7, 8	BO	Backward output

Eingangswahrheitstabelle

2pin Finput	1pin Binput	5,6pin Foutput	7,8pin Boutput
H	L	H	L
L	H	L	H
H	H	L	L
L	L	Open	Open

Beispiel

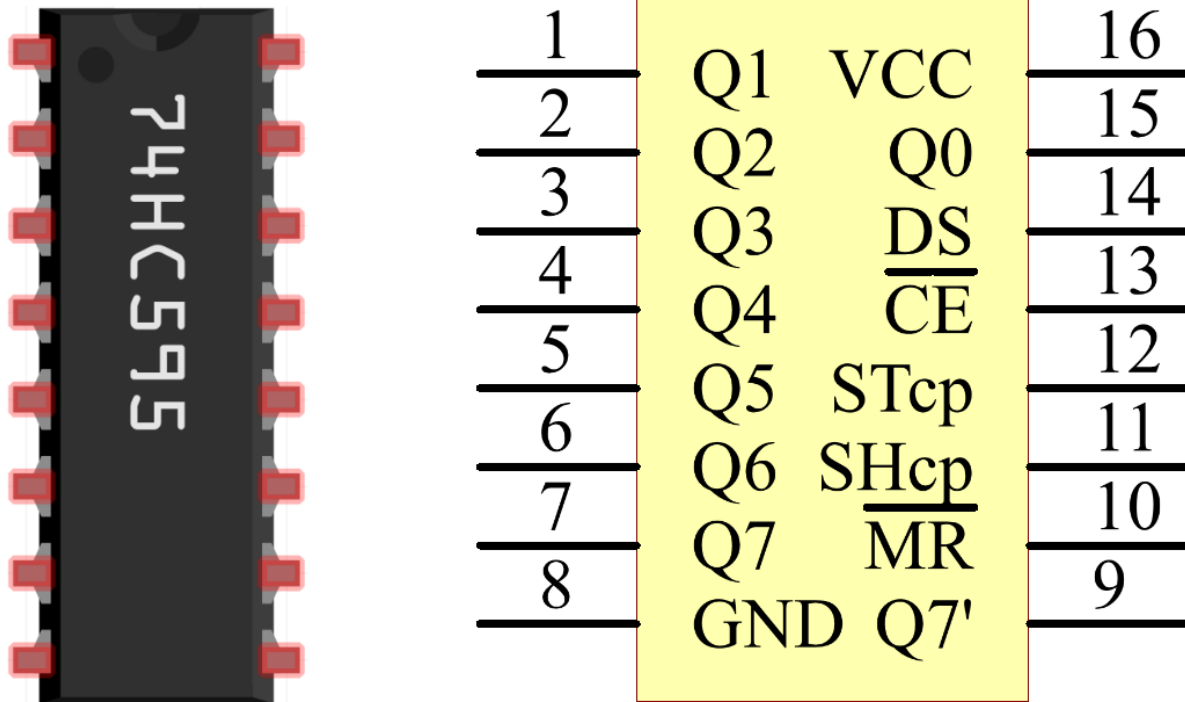
- *Motor* (Grundprojekt)
- *Wasserpumpe* (Grundprojekt)
- *Smarter Ventilator* (Spaßprojekt)
- *Pflanzenmonitor* (Spaßprojekt)

1.2.9 74HC595



Der 74HC595 besteht aus einem 8-Bit-Schieberegister und einem Speicherregister mit dreistufigen parallelen Ausgängen. Er wandelt serielle Eingaben in parallele Ausgaben um, sodass Sie IO-Ports eines Mikrocontrollers einsparen können. Wenn MR (Pin10) auf hohem Niveau und OE (Pin13) auf niedrigem Niveau ist, werden Daten bei steigender Flanke von SHcp eingegeben und gelangen durch die steigende Flanke von SHcp ins Speicherregister. Wenn die beiden Uhren miteinander verbunden sind, ist das Schieberegister immer einen Impuls früher als das Speicherregister. Im Speicherregister gibt es einen seriellen Schiebeeingangspin (Ds), einen seriellen Ausgangspin (Q) und einen asynchronen Reset-Knopf (niedriges Niveau). Das Speicherregister gibt einen Bus mit parallelen 8-Bit in drei Zuständen aus. Wenn OE aktiviert ist (niedriges Niveau), werden die Daten im Speicherregister auf den Bus ausgegeben.

•



Pins des 74HC595 und ihre Funktionen:

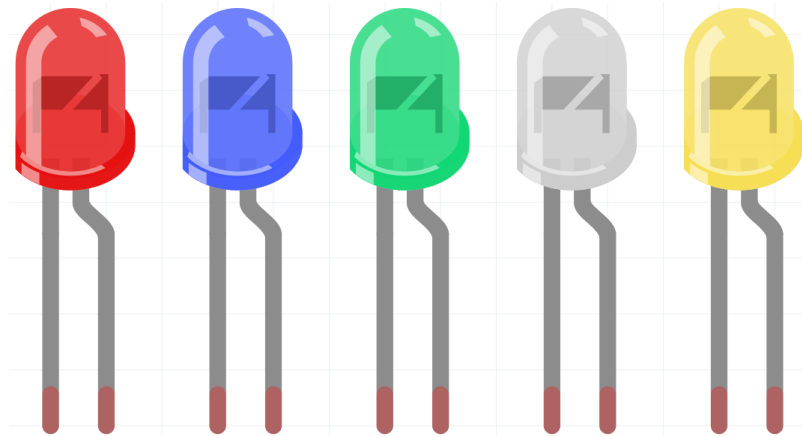
- **Q0-Q7**: 8-Bit parallele Datenausgangspins, können direkt 8 LEDs oder 8 Pins einer 7-Segment-Anzeige steuern.
- **Q7'**: Serieller Ausgangspin, verbunden mit DS eines anderen 74HC595, um mehrere 74HC595 in Serie zu schalten
- **MR**: Reset-Pin, aktiv bei niedrigem Pegel;
- **SHcp**: Zeitsequenzeingang des Schieberegisters. Bei steigender Flanke bewegen sich die Daten im Schieberegister sukzessive um ein Bit, d.h. Daten in Q1 bewegen sich nach Q2 usw. Während bei fallender Flanke die Daten im Schieberegister unverändert bleiben.
- **STcp**: Zeitsequenzeingang des Speicherregisters. Bei steigender Flanke bewegen sich die Daten im Schieberegister in das Speicherregister.
- **CE**: Ausgangsaktivierungspin, aktiv bei niedrigem Pegel.
- **DS**: Serieller Dateneingangspin
- **VCC**: Positive Versorgungsspannung.
- **GND**: Masse.

Beispiel

- [74HC595](#) (Grundprojekt)
- [Digitaler Würfel](#) (Spaßprojekt)

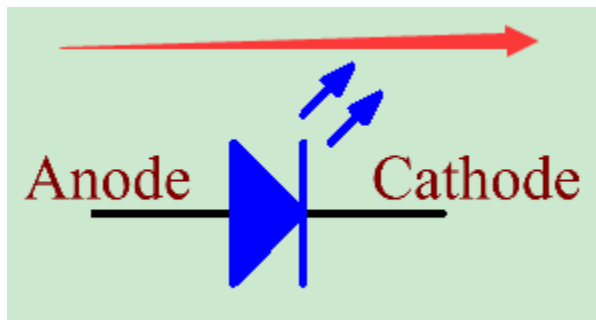
Anzeige

1.2.10 LED



Die Halbleiter-Leuchtdiode ist eine Art von Bauteil, das elektrische Energie durch PN-Übergänge in Lichtenergie umwandeln kann. Nach der Wellenlänge kann sie in Laserdiode, Infrarot-Leuchtdiode und sichtbare Leuchtdiode eingeteilt werden, die üblicherweise als Leuchtdiode (LED) bekannt ist.

Die Diode hat eine unidirektionale Leitfähigkeit, sodass der Stromfluss so erfolgt, wie der Pfeil im Schaltungssymbol der Abbildung anzeigt. Sie können nur die Anode mit positiver Energie und die Kathode mit negativer versorgen. So leuchtet die LED auf.



Eine LED hat zwei Pins. Der längere ist die Anode und der kürzere die Kathode. Achten Sie darauf, sie nicht verkehrt herum anzuschließen. In der LED gibt es einen festen Vorwärtsspannungsabfall, daher kann sie nicht direkt mit dem Stromkreis verbunden werden, da die Versorgungsspannung diesen Abfall übertreffen und die LED verbrennen lassen kann. Die Vorwärtsspannung der roten, gelben und grünen LED beträgt 1,8 V und die der weißen 2,6 V. Die meisten LEDs können einen maximalen Strom von 20 mA aushalten, daher müssen wir einen Strombegrenzungswiderstand in Reihe schalten.

Die Formel für den Widerstandswert lautet wie folgt:

$$R = (V_{\text{supply}} - V_D) / I$$

R steht für den Widerstandswert des Strombegrenzungswiderstands, **V_{supply}** für die Versorgungsspannung, **V_D** für den Spannungsabfall und **I** für den Arbeitsstrom der LED.

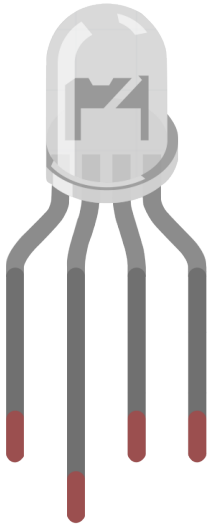
Hier ist die detaillierte Einführung für die LED: .

Beispiel

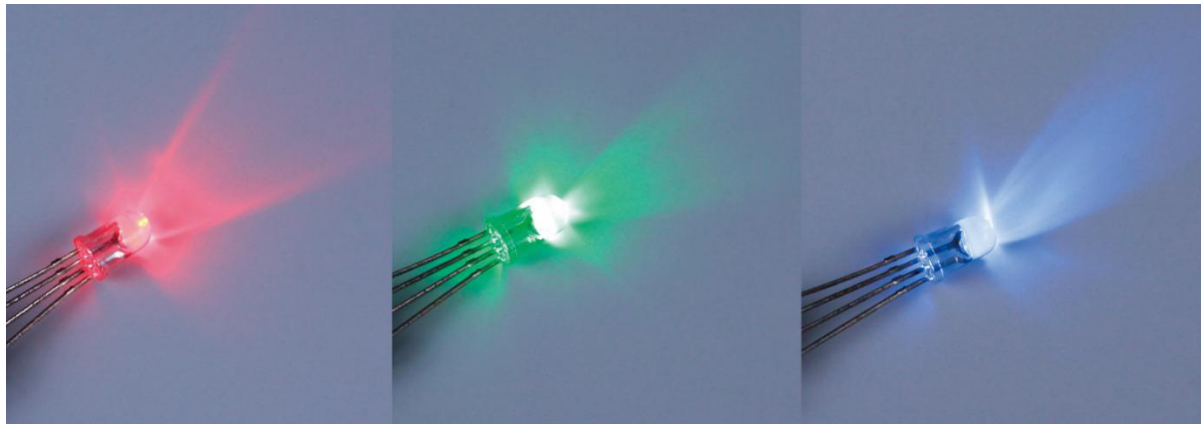
- [LED-Modul](#) (Grundprojekt)
- [Relais](#) (Grundprojekt)
- [Lichtempfindliches Array](#) (Spaßprojekt)

- *Smarter Ventilator* (Spaßprojekt)

1.2.11 RGB-LED

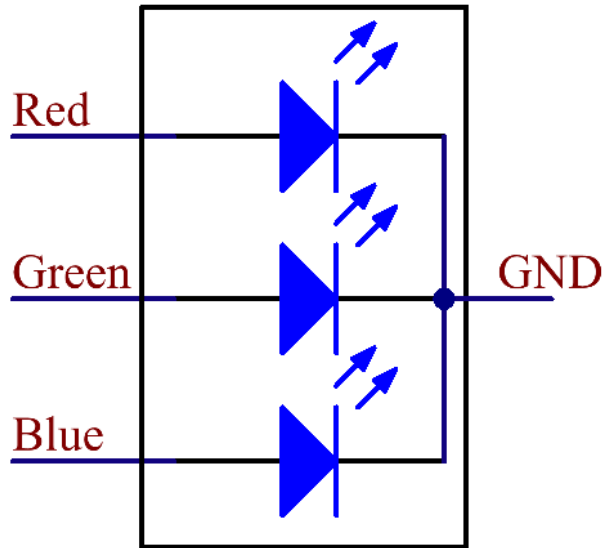


RGB-LEDs emittieren Licht in verschiedenen Farben. Eine RGB-LED verpackt drei LEDs in den Farben Rot, Grün und Blau in eine durchsichtige oder halbdurchsichtige Kunststoffhülle. Sie kann verschiedene Farben anzeigen, indem sie die Eingangsspannung der drei Pins ändert und diese überlagert, was laut Statistik 16.777.216 unterschiedliche Farben erzeugen kann.

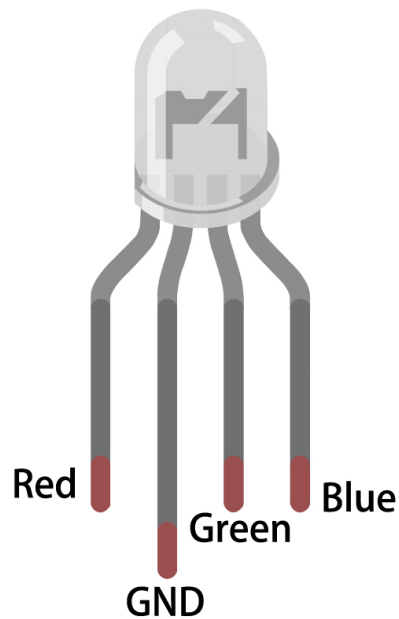


RGB-LEDs können in gemeinsame Anode und gemeinsame Kathode eingeteilt werden. In diesem Kit wird letztere verwendet. Die **gemeinsame Kathode**, oder CC, bedeutet, dass die Kathoden der drei LEDs verbunden sind. Nachdem Sie sie mit GND verbunden und die drei Pins eingesteckt haben, leuchtet die LED in der entsprechenden Farbe auf.

Ihr Schaltungssymbol wird wie folgt dargestellt.



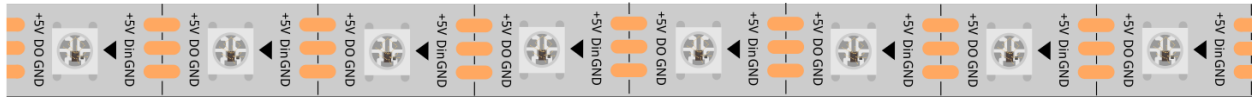
Eine RGB-LED hat 4 Pins: der längste ist GND; die anderen sind Rot, Grün und Blau. Berühren Sie ihre Kunststoffhülle und Sie werden eine Kerbe finden. Der Pin, der der Kerbe am nächsten ist, ist der erste Pin, markiert als Rot, dann GND, Grün und Blau der Reihe nach.



Beispiel

- *RGB-LED* (Grundprojekt)
- *HueDial* (Spaßprojekt)

1.2.12 WS2812 RGB 8 LEDs Streifen



Der WS2812 RGB 8 LEDs Streifen besteht aus 8 RGB-LEDs. Nur ein Pin ist erforderlich, um alle LEDs zu steuern. Jede RGB-LED hat einen WS2812-Chip, der unabhängig gesteuert werden kann. Es kann eine 256-stufige Helligkeitsanzeige realisieren und eine vollständige echte Farbanzeige von 16.777.216 Farben darstellen. Gleichzeitig enthält der Pixel einen intelligenten digitalen Schnittstellendatenrastsignalformungsverstärkerantriebsschaltkreis, und es ist ein Signalformungsschaltkreis integriert, um effektiv sicherzustellen, dass die Farbhöhe des Pixel-Punktlichts konsistent ist.

Er ist flexibel, kann angedockt, gebogen und nach Belieben geschnitten werden, und die Rückseite ist mit Klebeband ausgestattet, das nach Belieben auf unebenen Oberflächen befestigt werden kann und in engen Räumen installiert werden kann.

Merkmale

- Arbeitsspannung: DC5V
- IC: Ein IC steuert eine RGB-LED
- Verbrauch: 0,3 W pro LED
- Arbeitstemperatur: -15-50
- Farbe: Vollfarb-RGB
- RGB-Typ: 5050RGB (Integrierter IC WS2812B)
- Dicke des Lichtstreifens: 2mm
- Jede LED kann individuell gesteuert werden

WS2812B Einführung

•

WS2812B ist eine intelligente Steuerungs-LED-Lichtquelle, bei der der Steuerkreis und der RGB-Chip in einem 5050-Komponentenpaket integriert sind. Es beinhaltet intern einen intelligenten digitalen Port-Datenrast- und Signalaufbereitungsverstärkerantriebsschaltkreis. Ebenfalls enthalten ist ein präziser interner Oszillator und ein 12V programmierbarer Konstantstromsteuerungsteil, der effektiv sicherstellt, dass die Pixel-Punktlichtfarbhöhe konsistent ist.

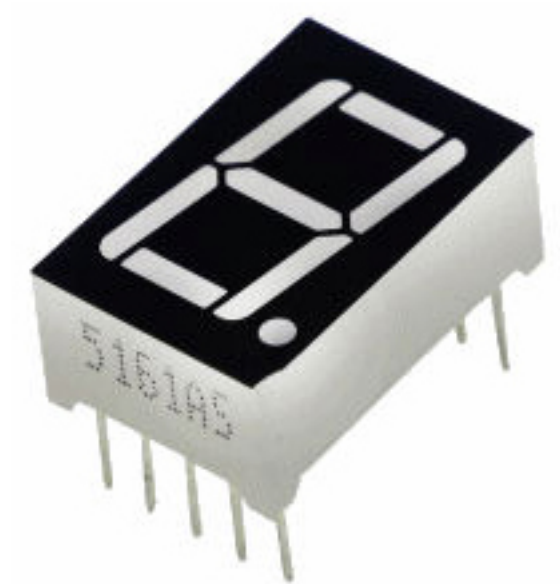
Das Datenübertragungsprotokoll verwendet den einzelnen NZR-Kommunikationsmodus. Nach dem Einschalten des Pixels setzt sich der DIN-Port zurück und empfängt Daten vom Controller, das erste Pixel sammelt die ersten 24 Bit Daten und sendet sie an den internen Datenrast, die anderen Daten, die durch den internen Signalaufbereitungsverstärkerkreis umgeformt wurden, werden über den DO-Port an das nächste Kaskadenpixel gesendet. Nach der Übertragung für jedes Pixel reduziert sich das Signal um 24 Bit. Pixel verwenden die Autoumformungsübertragungstechnologie, so dass die Anzahl der Pixelkaskaden nicht durch die Signalübertragung begrenzt wird, sondern nur von der Geschwindigkeit der Signalübertragung abhängt.

LED mit niedriger Ansteuerspannung, Umweltschutz und Energieeinsparung, hohe Helligkeit, große Streuwinkel, gute Konsistenz, geringe Leistung, lange Lebensdauer und andere Vorteile. Der in LED integrierte Steuerchip vereinfacht den Schaltkreis, reduziert das Volumen und erleichtert die Installation.

Beispiel

- *WS2812 RGB-LED-Streifen* (Grundprojekt)
- *CherryLight* (IoT-Projekt)

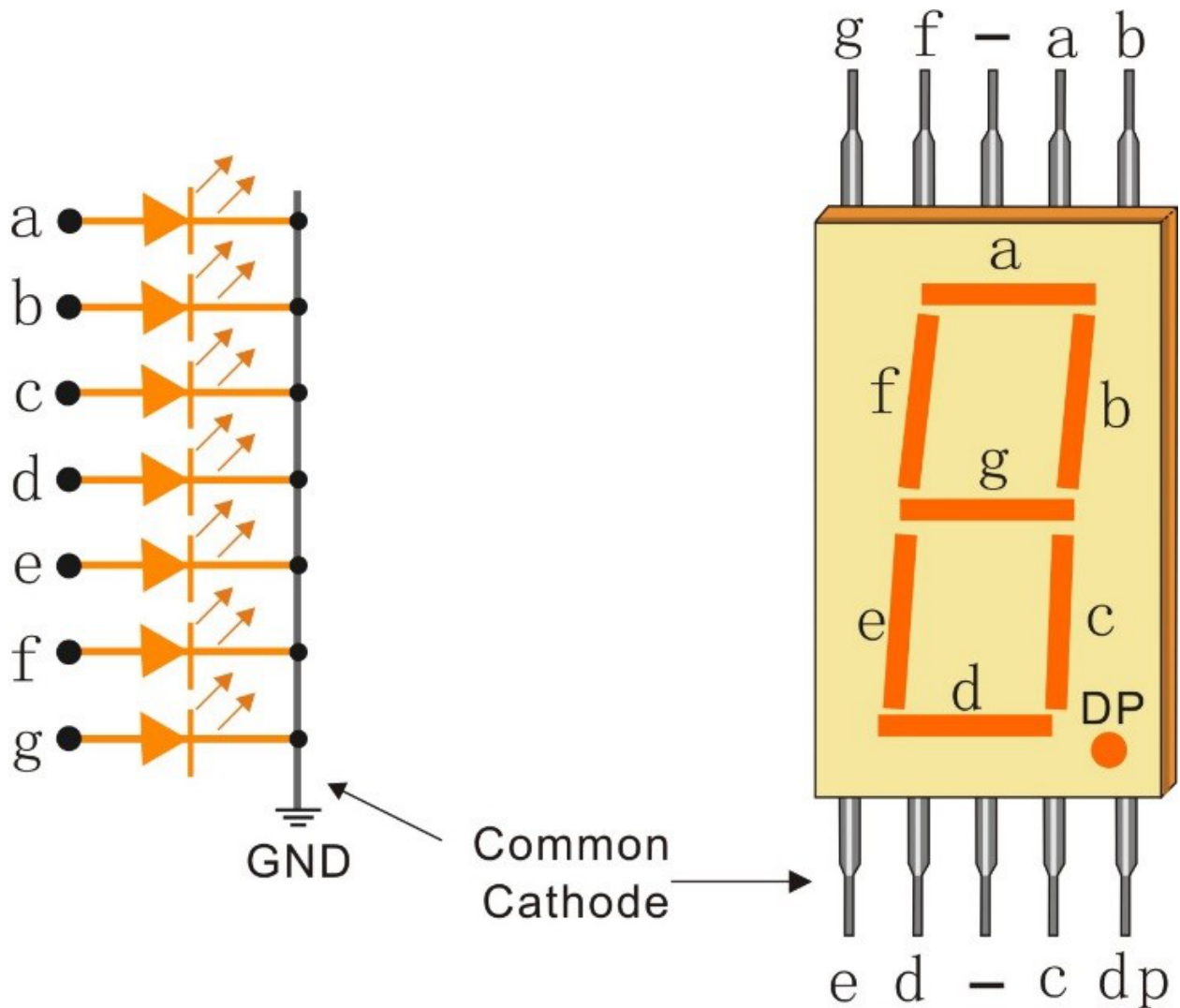
1.2.13 7-Segment-Anzeige



Eine 7-Segment-Anzeige ist eine 8-förmige Komponente, die 7 LEDs umfasst. Jede LED wird als Segment bezeichnet - wenn sie aktiviert wird, bildet jedes Segment einen Teil einer anzuzeigenden Zahl.

Es gibt zwei Arten von Pin-Verbindungen: Gemeinsame Kathode (CC) und Gemeinsame Anode (CA). Wie der Name schon sagt, hat eine CC-Anzeige alle Kathoden der 7 LEDs verbunden, während eine CA-Anzeige alle Anoden der 7 Segmente verbunden hat.

In diesem Kit verwenden wir die 7-Segment-Anzeige mit gemeinsamer Kathode. Hier ist das elektronische Symbol.



Jede der LEDs in der Anzeige erhält ein positionelles Segment mit einem ihrer Anschlusspins, der aus dem rechteckigen Kunststoffgehäuse herausgeführt wird. Diese LED-Pins sind von „a“ bis „g“ beschriftet und repräsentieren jede einzelne LED. Die anderen LED-Pins sind miteinander verbunden und bilden einen gemeinsamen Pin. Indem die entsprechenden Pins der LED-Segmente in einer bestimmten Reihenfolge in Vorwärtsrichtung polarisiert werden, leuchten einige Segmente auf und andere bleiben dunkel, sodass der entsprechende Charakter auf der Anzeige dargestellt wird.

Anzeigecodes

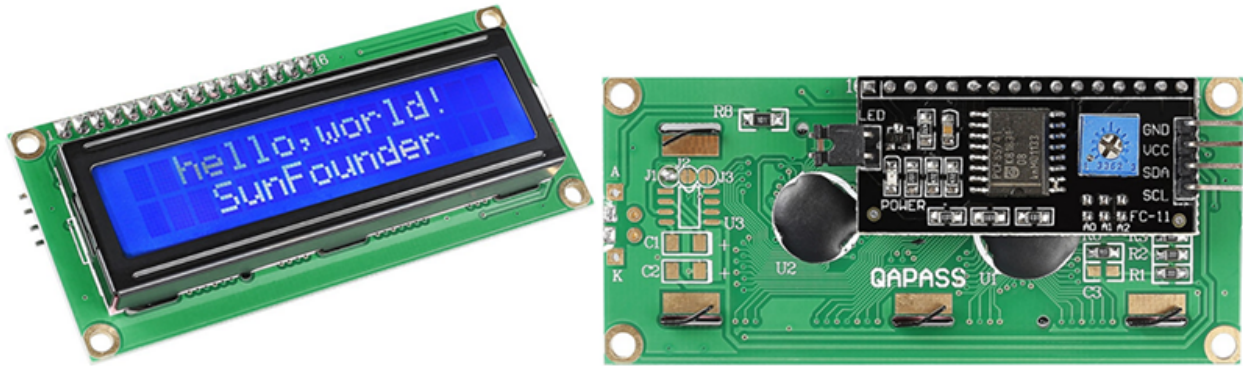
Um Ihnen zu helfen, zu verstehen, wie 7-Segment-Anzeigen (Gemeinsame Kathode) Zahlen darstellen, haben wir die folgende Tabelle gezeichnet. Zahlen sind die Zahlen 0-F, die auf der 7-Segment-Anzeige angezeigt werden; (DP) GFEDCBA bezieht sich auf das entsprechende LED-Set auf 0 oder 1, zum Beispiel bedeutet 00111111, dass DP und G auf 0 gesetzt sind, während die anderen auf 1 gesetzt sind. Daher wird die Zahl 0 auf der 7-Segment-Anzeige angezeigt, während HEX-Code der entsprechenden Hexadezimalzahl entspricht.

Numbers	Common Cathode		Numbers	Common Cathode	
	(DP)GFEDCBA	Hex Code		(DP)GFEDCBA A	Hex Code
0	00111111	0x3f	A	01110111	0x77
1	00000110	0x06	B	01111100	0x7c
2	01011011	0x5b	C	00111001	0x39
3	01001111	0x4f	D	01011110	0x5e
4	01100110	0x66	E	01111001	0x79
5	01101101	0x6d	F	01110001	0x71
6	01111101	0x7d			
7	00000111	0x07			
8	01111111	0x7f			
9	01101111	0x6f			

Beispiel

- *7-Segment-Anzeige* (Grundprojekt)
- *74HC595* (Grundprojekt)
- *Digitaler Würfel* (Spaßprojekt)

1.2.14 I2C LCD1602



- **GND:** Masse
- **VCC:** Spannungsversorgung, 5V.
- **SDA:** Serielle Datenleitung. Verbinden mit VCC über einen Pullup-Widerstand.
- **SCL:** Serielle Taktleitung. Verbinden mit VCC über einen Pullup-Widerstand.

Wie wir alle wissen, bereichern LCDs und andere Anzeigen zwar die Mensch-Maschine-Interaktion, teilen aber eine gemeinsame Schwäche. Wenn sie mit einem Controller verbunden sind, belegen sie mehrere IOs des Controllers, der nicht so viele externe Ports hat. Dies beschränkt auch andere Funktionen des Controllers.

Daher wurde das LCD1602 mit einem I2C-Modul entwickelt, um dieses Problem zu lösen. Das I2C-Modul hat einen eingebauten PCF8574 I2C-Chip, der I2C-Serien-Daten in parallele Daten für das LCD-Display umwandelt.

•

I2C-Adresse

Die Standardadresse ist grundsätzlich 0x27, in einigen Fällen kann sie 0x3F sein.

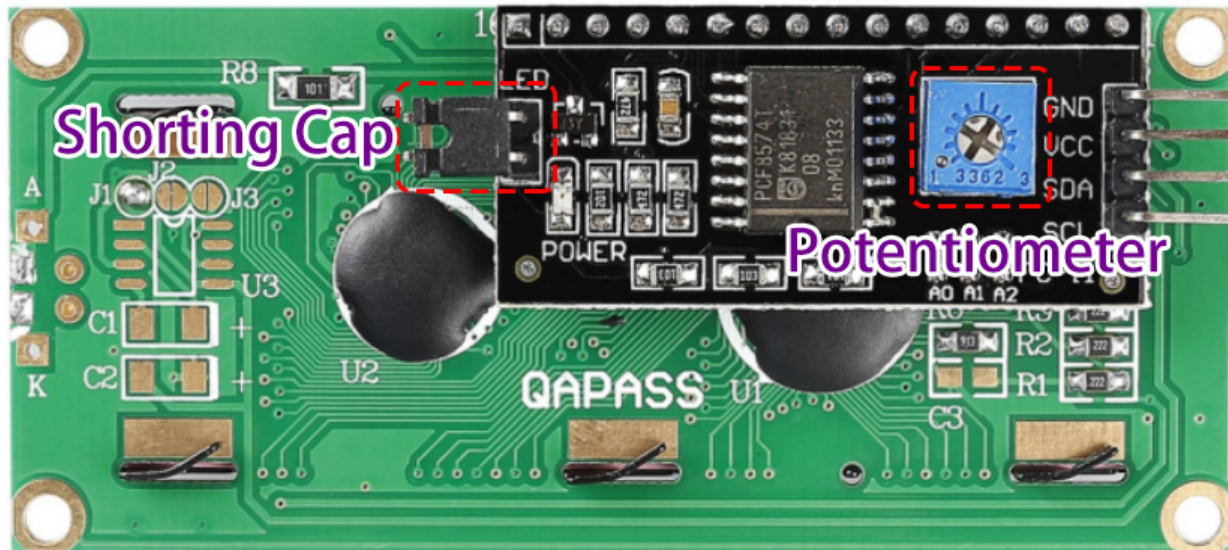
Nehmen wir die Standardadresse von 0x27 als Beispiel, kann die Geräteadresse geändert werden, indem die A0/A1/A2-Pads kurzgeschlossen werden; im Standardzustand ist A0/A1/A2 1, und wenn das Pad kurzgeschlossen wird, ist A0/A1/A2 0.

Slave Address

0	0	1	0	0	A2	A1	A0	
0	0	1	0	0	1	1	1	0x27
0	0	1	0	0	1	1	0	0x26
0	0	1	0	0	1	0	1	0x25
0	0	1	0	0	0	1	1	0x23
.....								
0	0	1	0	0	0	0	0	0x20

Hintergrundbeleuchtung/Kontrast

Die Hintergrundbeleuchtung kann durch einen Jumper aktiviert werden, entfernen Sie den Jumper, um die Hintergrundbeleuchtung zu deaktivieren. Das blaue Potentiometer auf der Rückseite wird verwendet, um den Kontrast (das Verhältnis der Helligkeit zwischen dem hellsten Weiß und dem dunkelsten Schwarz) einzustellen.



- **Kurzschlusskappe:** Die Hintergrundbeleuchtung kann durch diese Kappe aktiviert werden, entfernen Sie diese Kappe, um die Hintergrundbeleuchtung zu deaktivieren.

- **Potentiometer:** Es wird verwendet, um den Kontrast (die Klarheit des angezeigten Textes) einzustellen, der im Uhrzeigersinn erhöht und gegen den Uhrzeigersinn verringert wird.

Beispiel

- *I2C LCD1602* (Grundprojekt)
- *Ultraschall* (Grundprojekt)
- *Pflanzenmonitor* (Spaßprojekt)
- *SPIEL - Zahl Erraten* (Spaßprojekt)
- *Bluetooth-Nachrichtenbox* (IoT-Projekt)

1.2.15 OLED-Display-Modul



Einführung

Ein OLED-Display (Organic Light-Emitting Diode) ist ein Gerät, das Texte, Grafiken und Bilder auf einem dünnen und flexiblen Bildschirm anzeigen kann, indem es organische Materialien verwendet, die Licht abgeben, wenn ein elektrischer Strom angelegt wird.

Der Hauptvorteil eines OLED-Displays besteht darin, dass es sein eigenes Licht emittiert und keine andere Hintergrundlichtquelle benötigt. Dadurch haben OLED-Displays oft einen besseren Kontrast, eine bessere Helligkeit und Blickwinkel im Vergleich zu LCD-Displays.

Ein weiteres wichtiges Merkmal von OLED-Displays ist die tiefe Schwarzstufe. Da jeder Pixel in einem OLED-Display sein eigenes Licht emittiert, kann zur Erzeugung der schwarzen Farbe der einzelne Pixel ausgeschaltet werden.

Aufgrund des geringeren Stromverbrauchs (nur beleuchtete Pixel verbrauchen Strom) sind OLED-Displays auch bei batteriebetriebenen Geräten wie Smartwatches, Gesundheitstrackern und anderen Wearables beliebt.

Prinzip

Ein OLED-Display-Modul besteht aus einem OLED-Panel und einem OLED-Treiberchip, der auf der Rückseite des Moduls montiert ist. Das OLED-Panel besteht aus vielen kleinen Pixeln, die unterschiedliche Lichtfarben erzeugen können. Jeder Pixel besteht aus mehreren Schichten organischer Materialien, die zwischen zwei Elektroden (Anode und Kathode) eingeschlossen sind. Wenn elektrischer Strom durch die Elektroden fließt, emittieren die organischen Materialien Licht unterschiedlicher Wellenlängen, abhängig von ihrer Zusammensetzung.

Der OLED-Treiberchip ist ein Chip, der die Pixel des OLED-Panels über ein seriellen Kommunikationsprotokoll namens I2C (Inter-Integrated Circuit) steuern kann.

Der OLED-Treiberchip wandelt die Signale vom Arduino in Befehle für das OLED-Panel um. Der Arduino kann Daten an den OLED-Treiberchip senden, indem er eine Bibliothek verwendet, die das I2C-Protokoll steuern kann. Eine solche Bibliothek ist die Adafruit SSD1306-Bibliothek. Mit dieser Bibliothek können Sie das OLED-Displaymodul initialisieren, die Helligkeitsstufe einstellen, Texte, Grafiken oder Bilder ausgeben usw.

Beispiel

- *OLED* (Grundprojekt)
- *SPIEL - Pong* (Spaßprojekt)
- *WeatherTime Bildschirm* (IoT-Projekt)

Ton

1.2.16 Summer



Als eine Art elektronischer Summer mit integrierter Struktur werden Summer, die mit Gleichstrom versorgt werden, häufig in Computern, Druckern, Fotokopierern, Alarmanlagen, elektronischem Spielzeug, Automobil-Elektronikgeräten, Telefonen, Zeitgebern und anderen elektronischen Produkten oder Sprachgeräten eingesetzt.

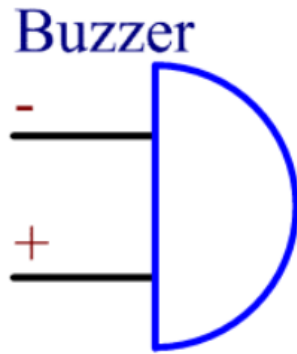
Summer können in aktive und passive Typen unterteilt werden (siehe das folgende Bild). Drehen Sie den Summer so, dass seine Pins nach oben zeigen, und der Summer mit einer grünen Platine ist ein passiver Summer, während derjenige, der mit schwarzem Klebeband umhüllt ist, ein aktiver ist.



Der Unterschied zwischen einem aktiven und einem passiven Summer:

Ein aktiver Summer hat eine eingebaute Oszillationsquelle, daher erzeugt er Töne, wenn er mit Strom versorgt wird. Ein passiver Summer hat jedoch keine solche Quelle, daher wird er nicht piepen, wenn Gleichstromsignale verwendet werden; stattdessen benötigen Sie Quadratwellen mit einer Frequenz zwischen 2K und 5K, um ihn zu betreiben. Der aktive Summer ist oft teurer als der passive aufgrund mehrerer eingebauter Oszillationsschaltungen.

Das Folgende ist das elektrische Symbol eines Summers. Er hat zwei Pins mit positiven und negativen Polen. Ein + auf der Oberfläche repräsentiert den Anodenpol und der andere ist der Kathodenpol.



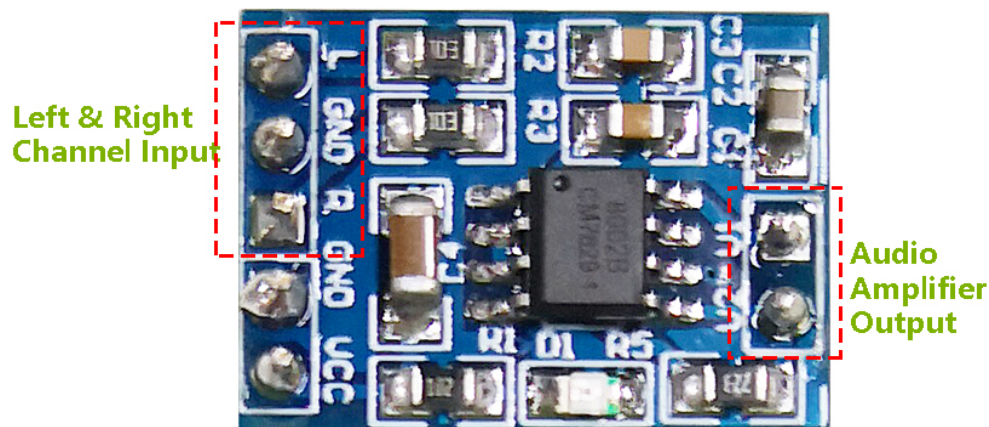
Sie können die Pins des Summers überprüfen, der längere ist der Anodenpol und der kürzere der Kathodenpol. Bitte verwechseln Sie diese nicht beim Anschließen, sonst wird der Summer keinen Ton erzeugen.

Beispiel

- *Aktiver Summer* (Grundprojekt)
- *Passiver Summer* (Grundprojekt)
- *Zugangskontrollsystem* (Spaßprojekt)

1.2.17 Audio-Modul und Lautsprecher

Audioverstärker-Modul



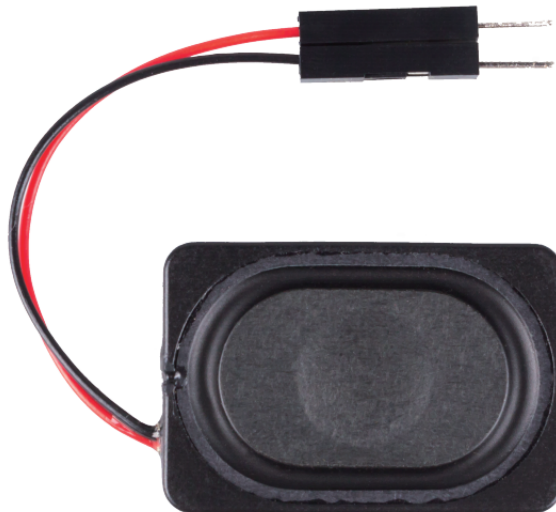
Das Audioverstärker-Modul enthält einen HXJ8002-Audioleistungsverstärker-Chip. Dieser Chip ist ein Leistungsverstärker mit niedriger Stromversorgung, der 3W durchschnittliche Audioleistung für eine 3Ω BTL-Last mit geringer

harmonischer Verzerrung (unter 10% Schwellenverzerrung bei 1KHz) von einer 5V DC-Stromquelle liefern kann. Dieser Chip kann Audiosignale ohne Kopplungskondensatoren oder Bootstrap-Kondensatoren verstärken.

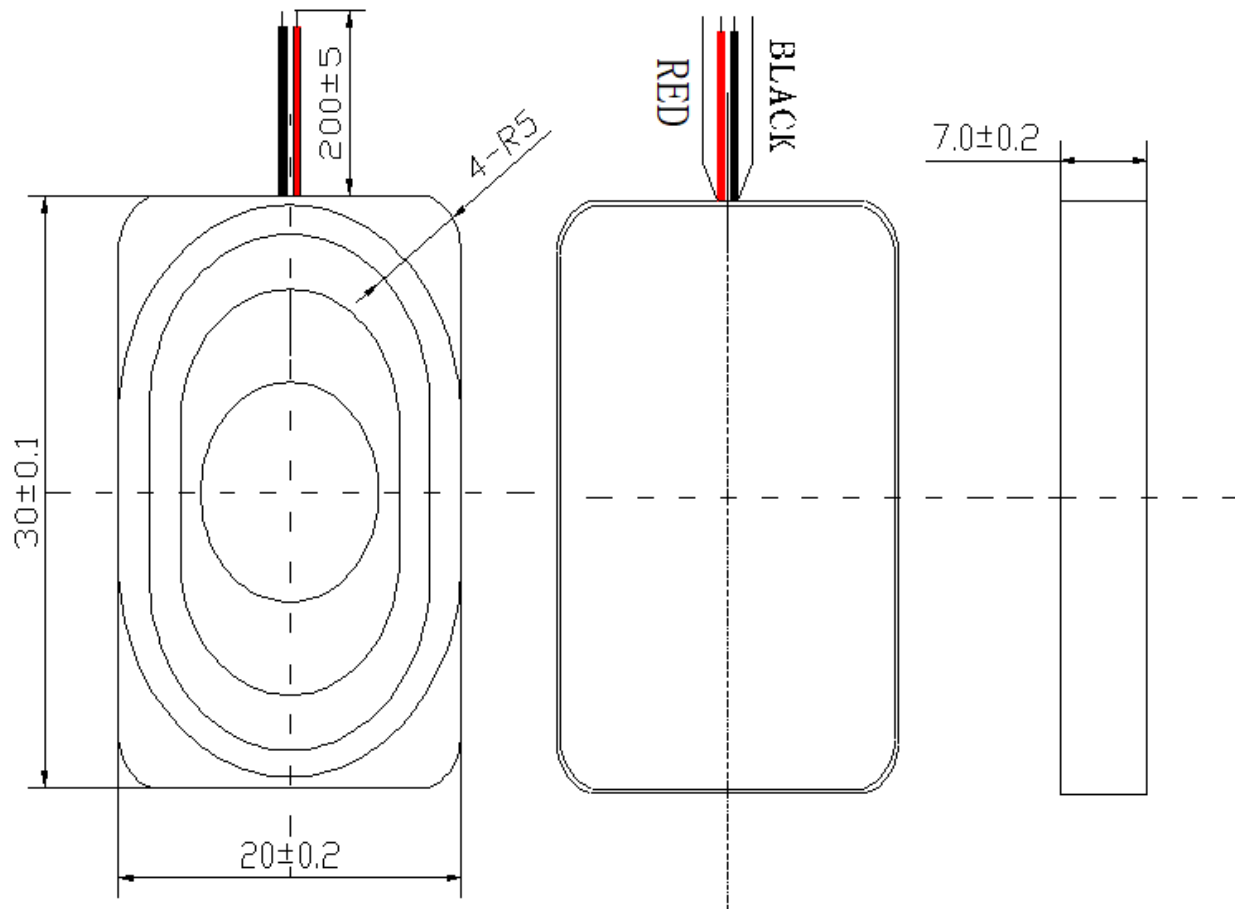
Das Modul kann mit einer 2,0V bis 5,5V DC Stromquelle mit 10mA Betriebsstrom (0,6uA für typischen Standby-Strom) betrieben werden und erzeugt einen kraftvollen verstärkten Klang in einen 3, 4 oder 8 Impedanzlautsprecher. Dieses Modul verfügt über eine verbesserte Pop- und Klickschaltung zur deutlichen Reduzierung des Übergangsgeräusches beim Ein- und Ausschalten. Die kleine Größe neben hoher Effizienz und niedriger Stromversorgung machen es weit verbreitet in tragbaren und batteriebetriebenen Projekten und Mikrocontrollern einsetzbar.

- **IC:** HXJ8002
- **Eingangsspannung:** 2V ~ 5.5V
- **Standby-Modus Strom:** 0.6uA (typischer Wert)
- **Ausgangsleistung:** 3W (3Ω Last), 2.5W (4Ω Last), 1.5W (8Ω Last)
- **Ausgangslautsprecherimpedanz:** 3Ω, 4Ω, 8Ω
- **Größe:** 19.8mm x 14.2mm

Lautsprecher



- **Größe:** 20x30x7mm
- **Impedanz:** 8ohm
- **Nennleistung:** 1.5W
- **Maximale Eingangsleistung:** 2.0W
- **Kabellänge:** 10cm



Die Größentabelle ist wie folgt:

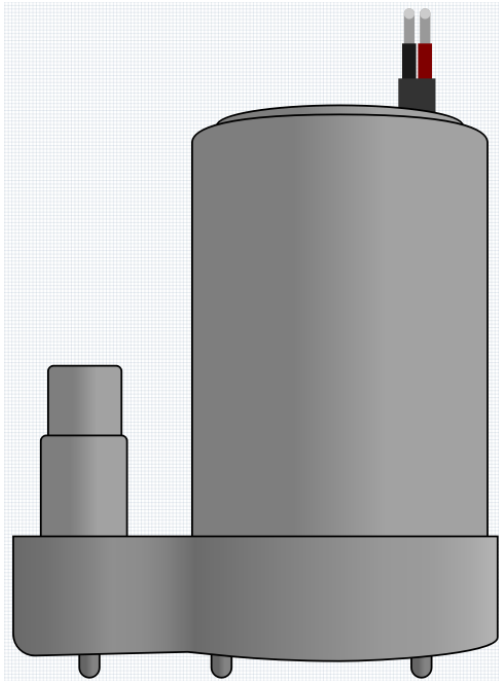
- 2030 Lautsprecher-Datenblatt

Beispiel

- *Audio-Modul und Lautsprecher* (Grundprojekt)
- *Willkommen* (Spaßprojekt)
- *Obstklavier* (Spaßprojekt)
- *Digital-Analog-Wandler (DAC)*

Treiber

1.2.18 DC-Wasserpumpe



Diese Pumpe funktioniert im Wesentlichen wie ein Gleichstrommotor, der mit einer Spannung von 3V und einem Strom von 100mA betrieben wird. Bei Stromzufuhr saugt die Pumpe Wasser von unten in ihr Kunststoffgehäuse ein und stößt es aus dem Auslassrohr aus. Sie muss immer in Wasser eingetaucht sein, um ordnungsgemäß zu funktionieren. Eine Umkehrung der Polarität macht sie nicht zu einem Wassereinzugsgerät; sie pumpt lediglich Wasser aus!

Sie eignet sich hervorragend für Anfänger, um ein Springbrunnen- oder Pflanzenbewässerungsprojekt mit dieser Tauchpumpe zu realisieren, da sie unglaublich benutzerfreundlich ist!

Merkmale

- **Spannungsbereich:** DC 3 ~ 4,5V
- **Betriebsstrom:** 120 ~ 180mA
- **Leistung:** 0,36 ~ 0,91W
- **Maximale Förderhöhe:** 0,35 ~ 0,55M
- **Maximale Fördermenge:** 80 ~ 100 L/H
- **Dauerbetriebsleben:** 100 Stunden
- **Wasserdichtigkeitsgrad:** IP68
- **Antriebsmodus:** DC, magnetischer Antrieb
- **Material:** Technischer Kunststoff
- **Außendurchmesser des Auslasses:** 7,8 mm
- **Innendurchmesser des Auslasses:** 6,5 mm
- Es ist eine Tauchpumpe und sollte auch so verwendet werden. Sie neigt dazu, zu überhitzen, weshalb es ein Risiko einer Überhitzung gibt, wenn Sie sie ungetaucht einschalten.
- Sie kommt mit einem 25 cm langen männlichen Kabel, das eine einfache Einführung in ein Steckbrett ermöglicht.

Beispiel

- *Wasserpumpe* (Grundprojekt)
- *Pflanzenmonitor* (Spaßprojekt)

1.2.19 Gleichstrommotor

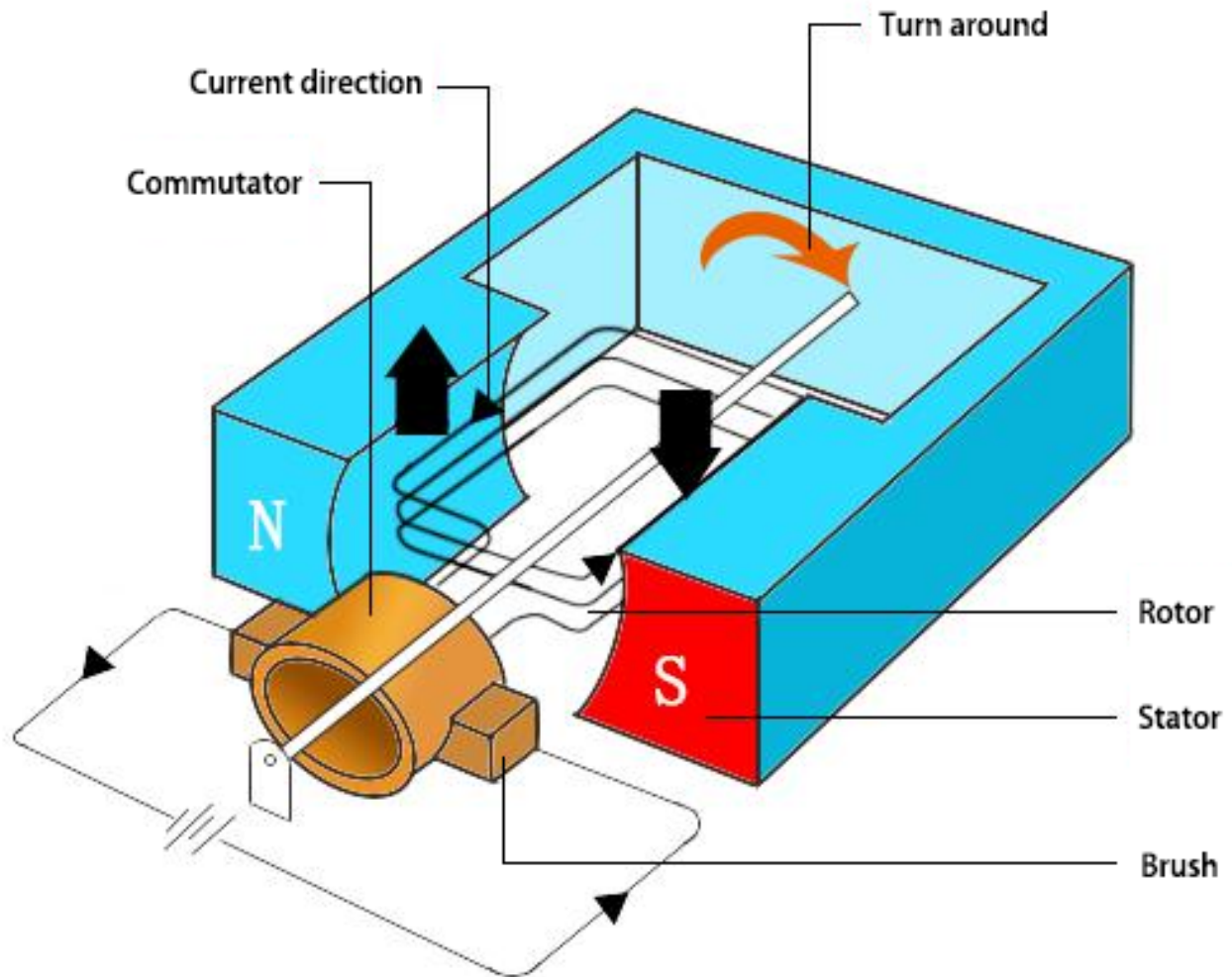


Dies ist ein 3V Gleichstrommotor. Wenn Sie an jeden der 2 Anschlüsse ein hohes und ein niedriges Niveau anlegen, wird er sich drehen.

- **Länge:** 25mm
- **Durchmesser:** 21mm
- **Wellendurchmesser:** 2mm
- **Wellenlänge:** 8mm
- **Spannung:** 3-6V
- **Strom:** 0,35-0,4A
- **Geschwindigkeit bei 3V:** 19000 U/min (Umdrehungen pro Minute)
- **Gewicht:** Ungefähr 14g (für eine Einheit)

Ein Gleichstrommotor ist ein kontinuierlicher Aktuator, der elektrische Energie in mechanische Energie umwandelt. Gleichstrommotoren sorgen dafür, dass rotierende Pumpen, Ventilatoren, Kompressoren, Impeller und andere Geräte durch kontinuierliche Winkelrotation funktionieren.

Ein Gleichstrommotor besteht aus zwei Teilen, dem festen Teil des Motors, der als **Stator** bezeichnet wird, und dem inneren Teil des Motors, der als **Rotor** (oder **Anker** eines Gleichstrommotors) bezeichnet wird und sich dreht, um Bewegung zu erzeugen. Der Schlüssel zur Erzeugung von Bewegung ist es, den Anker innerhalb des Magnetfeldes des Permanentmagneten zu positionieren (dessen Feld sich vom Nordpol zum Südpol erstreckt). Die Wechselwirkung des Magnetfeldes und der bewegten geladenen Partikel (der stromführende Draht erzeugt das Magnetfeld) erzeugt das Drehmoment, das den Anker dreht.



Der Strom fließt vom positiven Pol der Batterie durch den Schaltkreis, über die Kupferbürsten zum Kommutator und dann zum Anker. Aber wegen der zwei Lücken im Kommutator kehrt dieser Fluss auf halbem Weg jeder vollständigen Drehung um. Diese kontinuierliche Umkehr wandelt die Gleichstromleistung der Batterie im Wesentlichen in Wechselstrom um, wodurch der Anker das Drehmoment in der richtigen Richtung zur richtigen Zeit erfährt, um die Rotation aufrechtzuerhalten.

Beispiel

- *Motor* (Grundprojekt)
- *Smarter Ventilator* (Spaßprojekt)

1.2.20 Schrittmotor

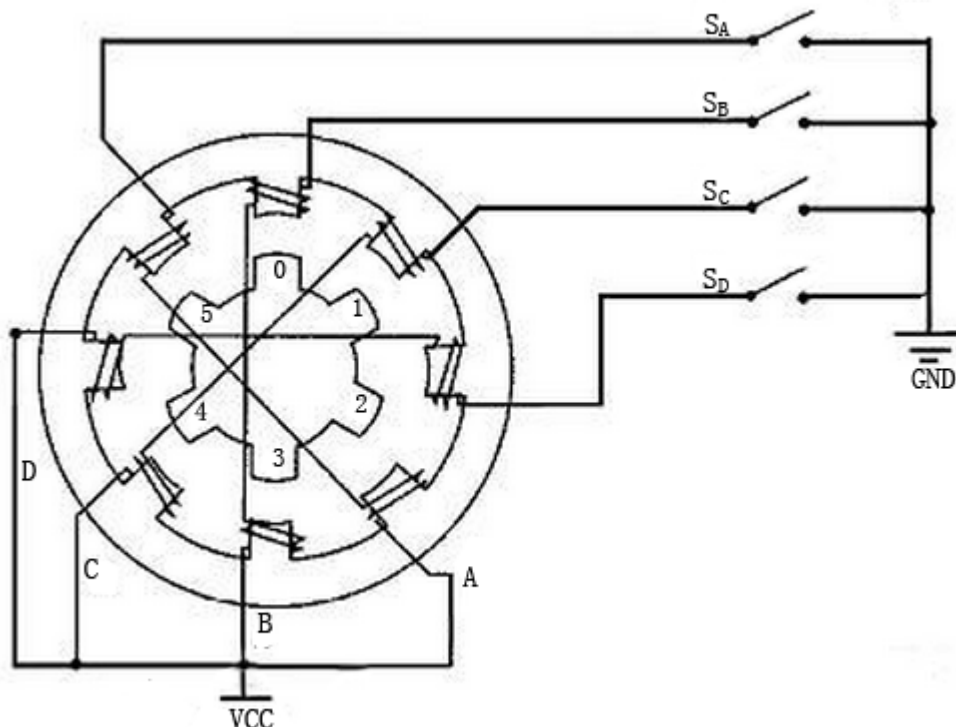


Schrittmotoren können aufgrund ihres einzigartigen Designs ohne Rückkopplungsmechanismen mit hoher Genauigkeit gesteuert werden. Die Welle eines Schrittmotors, die mit einer Serie von Magneten ausgestattet ist, wird von einer Serie elektromagnetischer Spulen gesteuert, die in einer bestimmten Abfolge positiv und negativ geladen sind und ihn präzise vorwärts oder rückwärts in kleinen „Schritten“ bewegen.

Prinzip

Es gibt zwei Arten von Schrittmotoren, Unipolare und Bipolare, und es ist sehr wichtig zu wissen, mit welchem Typ man arbeitet. In diesem Experiment werden wir einen Unipolaren Schrittmotor verwenden.

Der Schrittmotor ist ein vierphasiger Motor, der eine Unipolar-Gleichstromversorgung verwendet. Solange man alle Phasenwicklungen des Motors mit einer angemessenen zeitlichen Abfolge elektrifiziert, kann man ihn schrittweise drehen lassen. Das Schaltbild eines vierphasigen reaktiven Schrittmotors:



In der Abbildung, in der Mitte des Motors, ist ein Rotor - ein zahnradförmiger Permanentmagnet. Um den Rotor herum, von 0 bis 5, sind Zähne. Dann weiter außen gibt es 8 Magnetpole, wobei jeweils zwei gegenüberliegende durch Spulenwicklung verbunden sind. So bilden sie vier Paare von A bis D, was als Phase bezeichnet wird. Es hat vier Anschlussdrähte, die mit den Schaltern SA, SB, SC und SD verbunden werden. Daher sind die vier Phasen im Schaltkreis

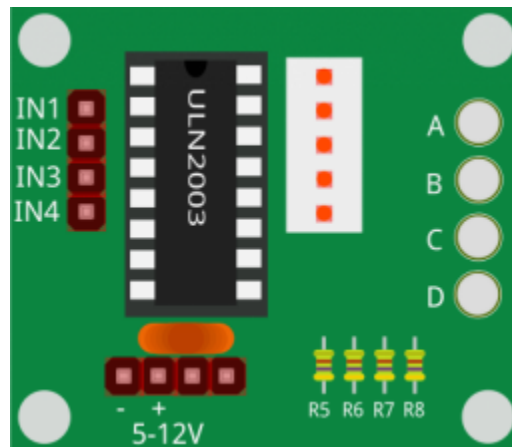
parallel und die zwei Magnetpole in einer Phase sind in Serie.

So funktioniert ein 4-Phasen-Schrittmotor:

Am Anfang ist Schalter SB eingeschaltet, Schalter SA, SC und SD sind ausgeschaltet, und B-Phasen-Magnetpole sind mit Zahn 0 und 3 des Rotors ausgerichtet. Gleichzeitig erzeugen Zahn 1 und 4 versetzte Zähne mit den C- und D-Phasenpolen. Zahn 2 und 5 erzeugen versetzte Zähne mit den D- und A-Phasenpolen. Wenn Schalter SC eingeschaltet ist, Schalter SB, SA und SD sind ausgeschaltet, dreht sich der Rotor unter dem Magnetfeld der C-Phasenwicklung und dem zwischen Zahn 1 und 4. Dann richten sich Zahn 1 und 4 mit den Magnetpolen der C-Phasenwicklung aus. Während Zahn 0 und 3 versetzte Zähne mit den A- und B-Phasenpolen erzeugen und Zahn 2 und 5 versetzte Zähne mit den Magnetpolen der A- und D-Phasenpolen. Die ähnliche Situation setzt sich fort. Erregen Sie die A-, B-, C- und D-Phasen der Reihe nach, und der Rotor dreht sich in der Reihenfolge A, B, C und D.

Der vierphasige Schrittmotor hat drei Betriebsmodi: Einzelschritt-Vierer-Schritt, Doppelschritt-Vierer-Schritt und Achterschritt. Der Schrittwinkel für den Einzelschritt-Vierer-Schritt und Doppelschritt-Vierer-Schritt sind gleich, aber das Antriebsmoment für den Einzelschritt-Vierer-Schritt ist kleiner. Der Schrittwinkel des Achterschritts ist halb so groß wie der des Einzelschritt-Vierer-Schritts und Doppelschritt-Vierer-Schritts. Daher kann der Achterschritt-Betriebsmodus ein hohes Antriebsmoment beibehalten und die Steuerungsgenauigkeit verbessern. In diesem Experiment lassen wir den Schrittmotor im Achterschritt-Modus arbeiten.

ULN2003-Modul



Um den Motor im Schaltkreis zu verwenden, muss eine Treiberplatine verwendet werden. Schrittmotor-Treiber-ULN2003 ist ein 7-Kanal-Inverterschaltkreis. Das heißt, wenn das Eingangsende auf hohem Niveau ist, ist das Ausgangsende des ULN2003 auf niedrigem Niveau und umgekehrt. Wenn wir hohes Niveau an IN1 liefern und niedriges Niveau an IN2, IN3 und IN4, dann ist das Ausgangsende OUT1 auf niedrigem Niveau und alle anderen Ausgangsenden sind auf hohem Niveau. So leuchtet D1 auf, Schalter SA ist eingeschaltet und der Schrittmotor dreht sich einen Schritt. Der ähnliche Fall wiederholt sich immer wieder. Geben Sie dem Schrittmotor also eine bestimmte zeitliche Abfolge, und er wird schrittweise rotieren. Der ULN2003 wird hier verwendet, um bestimmte zeitliche Abfolgen für den Schrittmotor bereitzustellen.

Beispiel

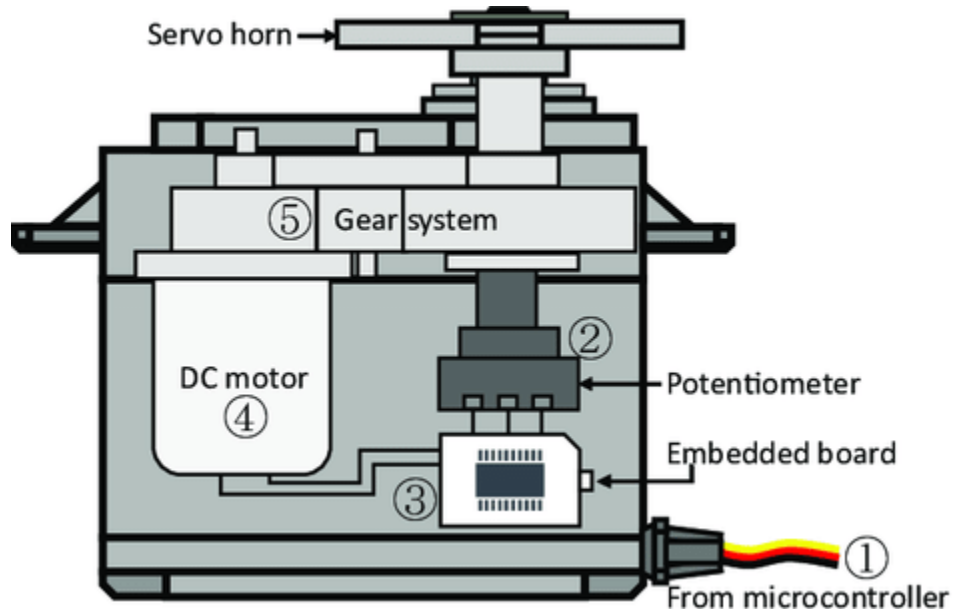
- [Schrittmotor](#) (Grundprojekt)
- [Zugangskontrollsystem](#) (Spaßprojekt)

1.2.21 Servomotor

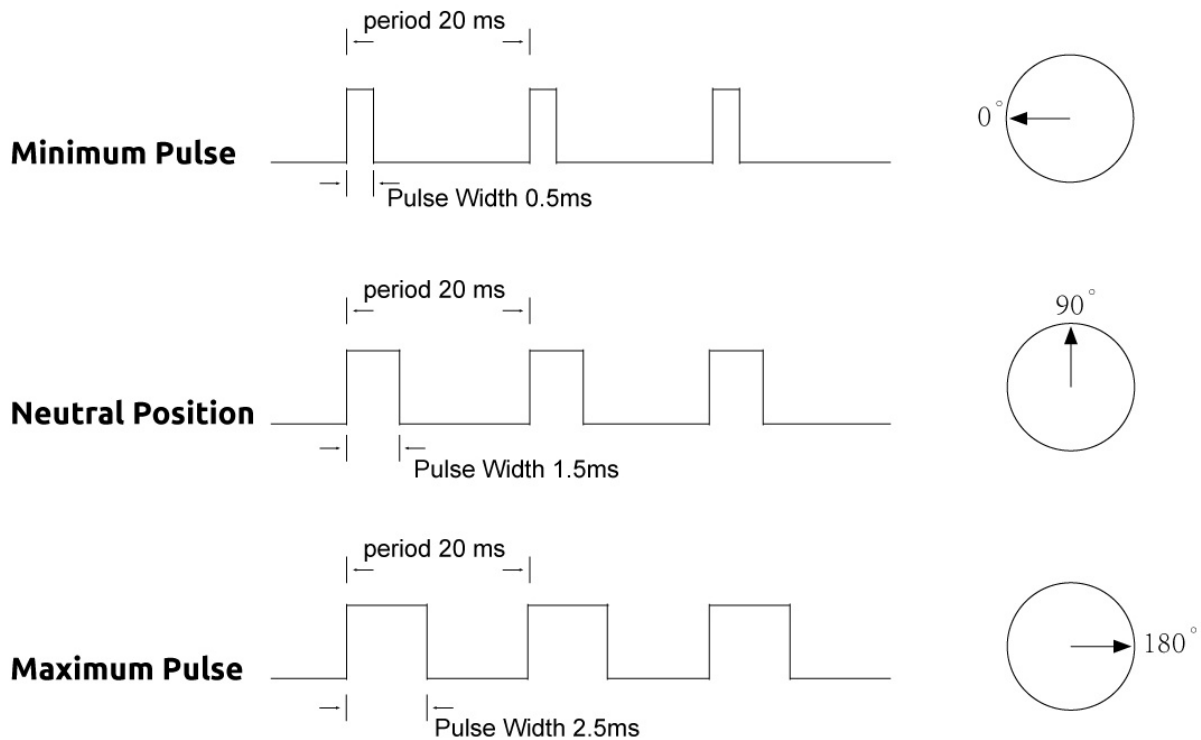


Ein Servomotor besteht im Allgemeinen aus den folgenden Teilen: Gehäuse, Welle, Getriebesystem, Potentiometer, Gleichstrommotor und eingebettetes Board.

So funktioniert es: Der Mikrocontroller sendet PWM-Signale an den Servo, und dann empfängt das eingebettete Board im Servo die Signale über den Signaleingang und steuert den Motor im Inneren zur Drehung. Infolgedessen treibt der Motor das Getriebesystem an und motiviert dann die Welle nach der Verzögerung. Die Welle und das Potentiometer des Servos sind miteinander verbunden. Wenn sich die Welle dreht, treibt sie das Potentiometer an, sodass das Potentiometer ein Spannungssignal an das eingebettete Board ausgibt. Dann bestimmt das Board die Drehrichtung und -geschwindigkeit basierend auf der aktuellen Position, sodass es genau an der definierten Position stoppen und dort halten kann.



Der Winkel wird durch die Dauer eines Impulses bestimmt, der auf den Steuerdraht aufgebracht wird. Dies wird als Pulsweitenmodulation bezeichnet. Der Servo erwartet alle 20 ms einen Impuls. Die Länge des Impulses bestimmt, wie weit der Motor dreht. Zum Beispiel bewirkt ein 1,5 ms langer Impuls, dass der Motor in die 90-Grad-Position (Neutrale Position) dreht. Wenn ein Impuls an einen Servo gesendet wird, der kürzer als 1,5 ms ist, dreht sich der Servo in eine Position und hält seine Ausgangswelle um eine bestimmte Anzahl von Grad gegen den Uhrzeigersinn vom Neutralpunkt. Wenn der Impuls breiter als 1,5 ms ist, tritt das Gegenteil ein. Die minimale Breite und die maximale Breite des Impulses, der den Servo anweist, sich in eine gültige Position zu drehen, sind Funktionen jedes Servos. Im Allgemeinen wird der minimale Impuls etwa 0,5 ms breit und der maximale Impuls 2,5 ms breit sein.



Beispiel

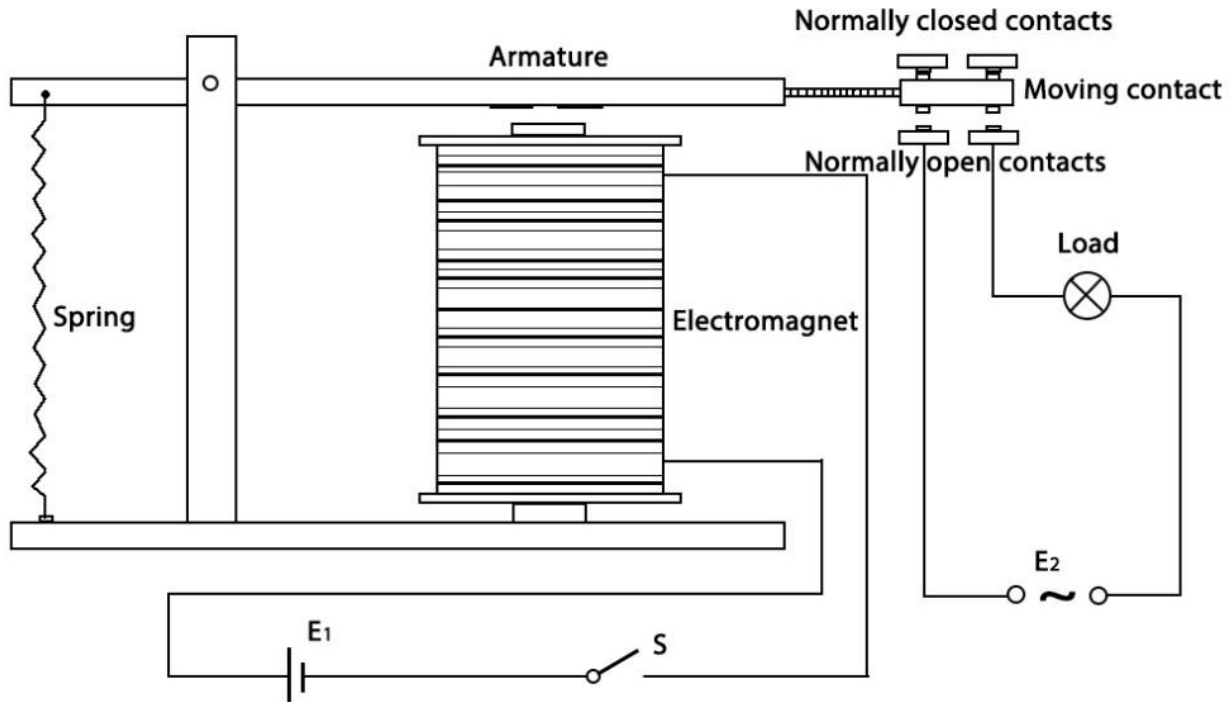
- *Servo* (Grundprojekt)
- *Smarter Mülleimer* (Spaßprojekt)

1.2.22 Relais



Wie wir vielleicht wissen, ist ein Relais ein Gerät, das dazu dient, eine Verbindung zwischen zwei oder mehr Punkten oder Geräten als Reaktion auf das angelegte Eingangssignal herzustellen. Mit anderen Worten, Relais bieten eine Isolierung zwischen dem Controller und dem Gerät, da Geräte sowohl mit Wechselstrom als auch mit Gleichstrom betrieben werden können. Sie erhalten jedoch Signale von einem Mikrocontroller, der mit Gleichstrom arbeitet, und benötigen daher ein Relais, um die Lücke zu überbrücken. Relais sind äußerst nützlich, wenn Sie eine große Menge an Strom oder Spannung mit einem kleinen elektrischen Signal steuern müssen.

Es gibt 5 Teile in jedem Relais:



Electromagnet - Er besteht aus einem Eisenkern, der von Spulen umwickelt ist. Wenn Strom hindurchfließt, wird er magnetisch. Daher wird er Elektromagnet genannt.

Armature - Der bewegliche magnetische Streifen wird als Anker bezeichnet. Wenn Strom durch ihn fließt, wird die Spule energisiert und erzeugt ein Magnetfeld, das verwendet wird, um die normalerweise offenen (N/O) oder normalerweise geschlossenen (N/C) Kontakte herzustellen oder zu unterbrechen. Und der Anker kann mit Gleichstrom (DC) sowie Wechselstrom (AC) bewegt werden.

Spring - Wenn kein Strom durch die Spule am Elektromagneten fließt, zieht die Feder den Anker weg, sodass der Stromkreis nicht geschlossen werden kann.

Satz elektrischer **contacts** - Es gibt zwei Kontaktstellen:

- Normalerweise offen - verbunden, wenn das Relais aktiviert ist, und getrennt, wenn es inaktiv ist.
- Normalerweise geschlossen - nicht verbunden, wenn das Relais aktiviert ist, und verbunden, wenn es inaktiv ist.

Molded frame - Relais sind zum Schutz mit Kunststoff überzogen.

Das Funktionsprinzip eines Relais ist einfach. Wenn Strom an das Relais geliefert wird, beginnt der Strom durch die Steuerspule zu fließen; als Ergebnis beginnt der Elektromagnet sich zu aktivieren. Dann wird der Anker zur Spule gezogen und zieht den beweglichen Kontakt nach unten, wodurch er sich mit den normalerweise offenen Kontakten verbindet. So wird der Stromkreis mit der Last aktiviert. Das Unterbrechen des Stromkreises wäre ein ähnlicher Fall, da der bewegliche Kontakt unter der Kraft der Feder zu den normalerweise geschlossenen Kontakten gezogen wird. Auf diese Weise kann das Ein- und Ausschalten des Relais den Zustand eines Lastkreises steuern.

Beispiel

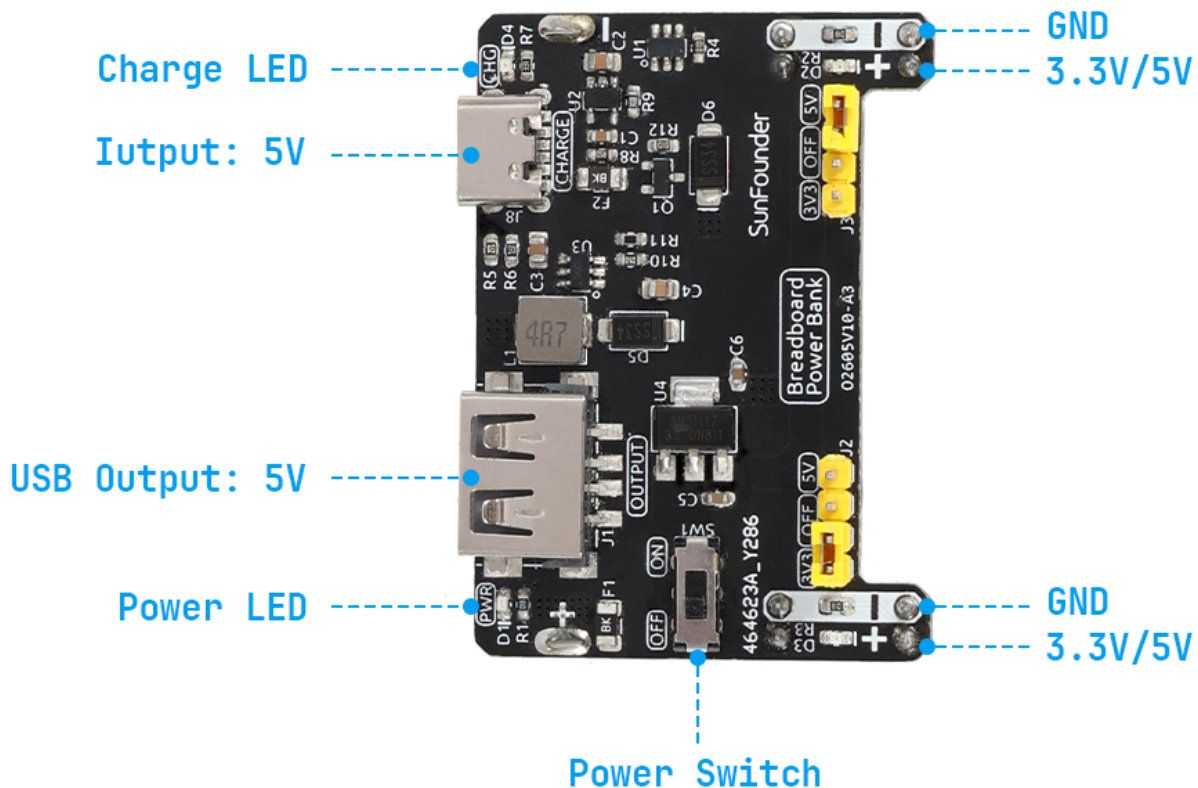
- *Relais* (Grundprojekt)

1.2.23 Stromversorgungsmodul

Bemerkung: Um den Akku des *Stromversorgungsmodul* zu schonen, laden Sie ihn bitte vor dem ersten Gebrauch vollständig auf.

Wenn wir einen großen Strom benötigen, um eine Komponente zu betreiben, kann dies die normale Arbeit des Arduino UNO-Boards stark beeinträchtigen. Daher versorgen wir die Komponente separat mit diesem Modul, um sie sicher und stabil laufen zu lassen.

Sie können es einfach in das Steckbrett stecken, um Strom zu liefern. Es bietet eine Spannung von 3,3V und 5V, und Sie können entweder über eine mitgelieferte Jumperkappe verbinden.



Merkmale und Spezifikationen

- Stromversorgungseingang: USB Typ-C, 5V
- Ausgangsspannung: 5V, 3.3V (einstellbar über Jumper. 0V, 3.3V und 5V Konfiguration)
- Ausgangsstrom: 5V/1,5A, 3,3V/1A
- Ein-Aus-Schalter verfügbar
- Zwei unabhängige Kanäle
- Onboard-Berg-Männchen-Header für GND, 5V, 3.3V Ausgang
- USB (Typ-C) Ausgang verfügbar
- USB (Typ-A) Eingang verfügbar
- An Bord LED: Ladeanzeige CHG, Betriebsanzeige PWR
- Akku: 3,7V 14500 Lithium-Ionen-Akku, 500mAh

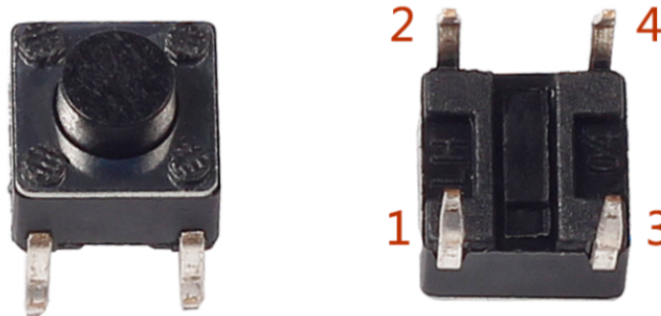
- Abmessungen: 52mm x 32mm (L x B)

Beispiel

- *Motor* (Grundprojekt)
- *Wasserpumpe* (Grundprojekt)
- *Schrittmotor* (Grundprojekt)
- *Smarter Ventilator* (Spaßprojekt)
- *Pflanzenmonitor* (Spaßprojekt)
- *Zugangskontrollsystem* (Spaßprojekt)
- *SPIEL - Pong* (Spaßprojekt)

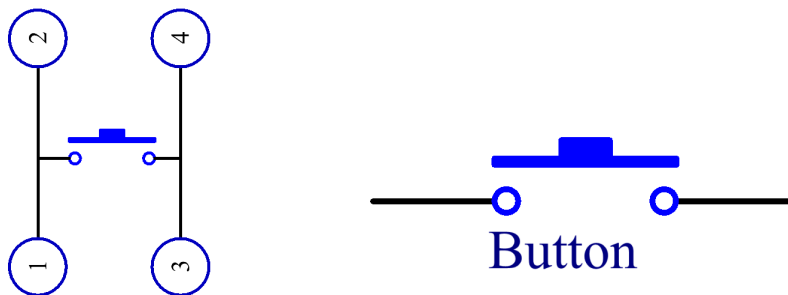
Steuerung

1.2.24 Knopf

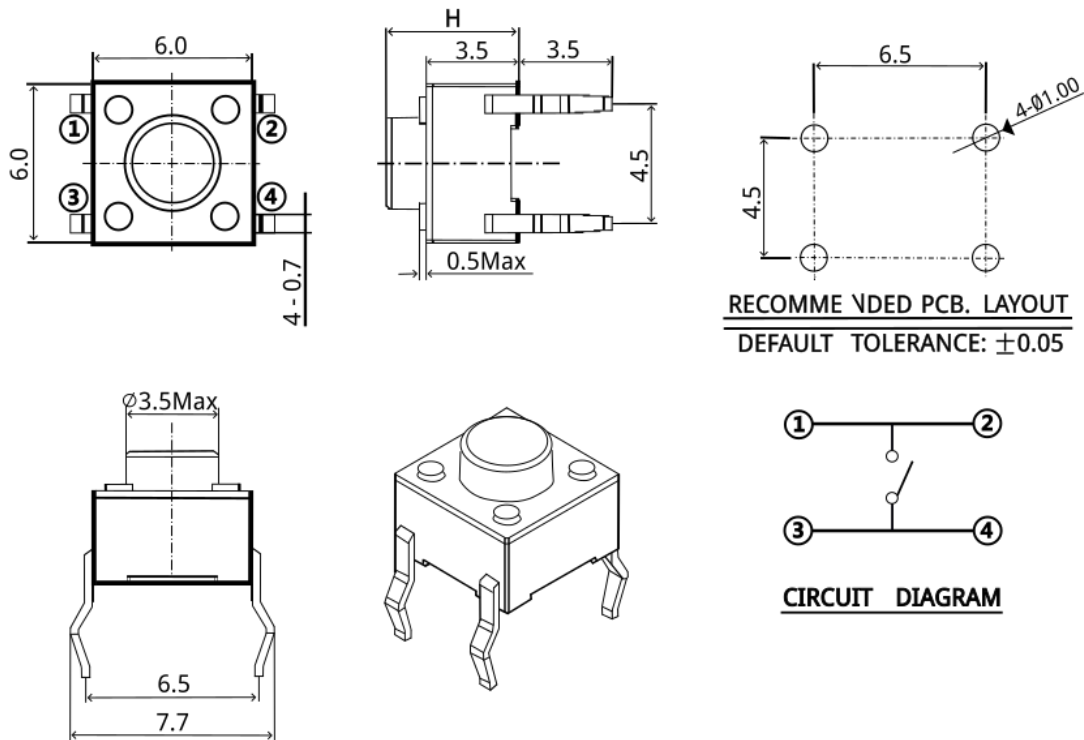


Knöpfe sind eine gängige Komponente zur Steuerung elektronischer Geräte. Sie werden meist als Schalter verwendet, um Stromkreise zu schließen oder zu unterbrechen. Obwohl es Knöpfe in verschiedenen Größen und Formen gibt, wird hier ein 6mm Mini-Knopf verwendet, wie in den folgenden Bildern gezeigt. Pin 1 ist mit Pin 2 und Pin 3 mit Pin 4 verbunden. Sie müssen also entweder Pin 1 mit Pin 3 oder Pin 2 mit Pin 4 verbinden.

Das Folgende ist die innere Struktur eines Knopfes. Das Symbol unten rechts wird üblicherweise verwendet, um einen Knopf in Schaltungen darzustellen.



Da Pin 1 mit Pin 2 verbunden ist und Pin 3 mit Pin 4, werden beim Drücken des Knopfes die 4 Pins verbunden, wodurch der Stromkreis geschlossen wird.



In diesem Kit bieten wir zwei Arten von Knöpfen an. Der zuvor erwähnte ist ein kleiner Knopf, und es gibt auch einen großen Knopf. Sie haben dasselbe Prinzip, unterscheiden sich jedoch in der Größe.

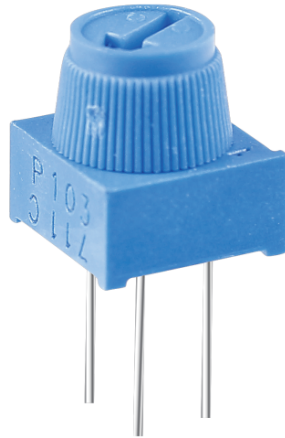


Beispiel

- *Taster* (Grundprojekt)
- *Digitaler Würfel* (Spaßprojekt)
- *Smarter Ventilator* (Spaßprojekt)
- *SPIEL - Pong* (Spaßprojekt)

- *Cloud-Anrufsystem mit MQTT* (IoT-Projekt)
- *Tastatursteuerung*

1.2.25 Potentiometer

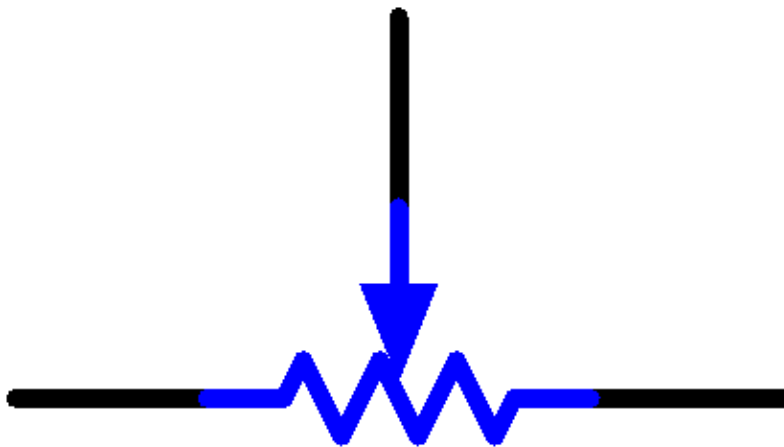


Ein Potentiometer ist ebenfalls eine Widerstandskomponente mit 3 Anschlüssen, und sein Widerstandswert kann gemäß einer regelmäßigen Variation eingestellt werden.

Potentiometer gibt es in verschiedenen Formen, Größen und Werten, aber sie haben alle folgende Gemeinsamkeiten:

- Sie haben drei Anschlüsse (oder Verbindungspunkte).
- Sie haben einen Knopf, eine Schraube oder einen Schieber, der bewegt werden kann, um den Widerstand zwischen dem mittleren Anschluss und einem der beiden äußeren Anschlüsse zu variieren.
- Der Widerstand zwischen dem mittleren Anschluss und einem der beiden äußeren Anschlüsse variiert von 0 bis zum maximalen Widerstand des Potentiometers, wenn der Knopf, die Schraube oder der Schieber bewegt wird.

Hier ist das Schaltungssymbol eines Potentiometers.



Die Funktionen des Potentiometers im Schaltkreis sind wie folgt:

1. Dient als Spannungsteiler

Ein Potentiometer ist ein kontinuierlich einstellbarer Widerstand. Wenn Sie die Welle oder den Schiebegriff des Potentiometers einstellen, wird der bewegliche Kontakt auf dem Widerstand gleiten. In

diesem Moment kann eine Spannung ausgegeben werden, abhängig von der Spannung, die an das Potentiometer angelegt wird, und dem Winkel, den der bewegliche Arm gedreht hat, oder dem Weg, den er zurückgelegt hat.

2. Dient als Rheostat

Wenn das Potentiometer als Rheostat verwendet wird, verbinden Sie den mittleren Pin und einen der anderen 2 Pins im Schaltkreis. So können Sie einen stufenlos und kontinuierlich veränderten Widerstandswert innerhalb des Weges des beweglichen Kontakts erhalten.

3. Dient als Stromregler

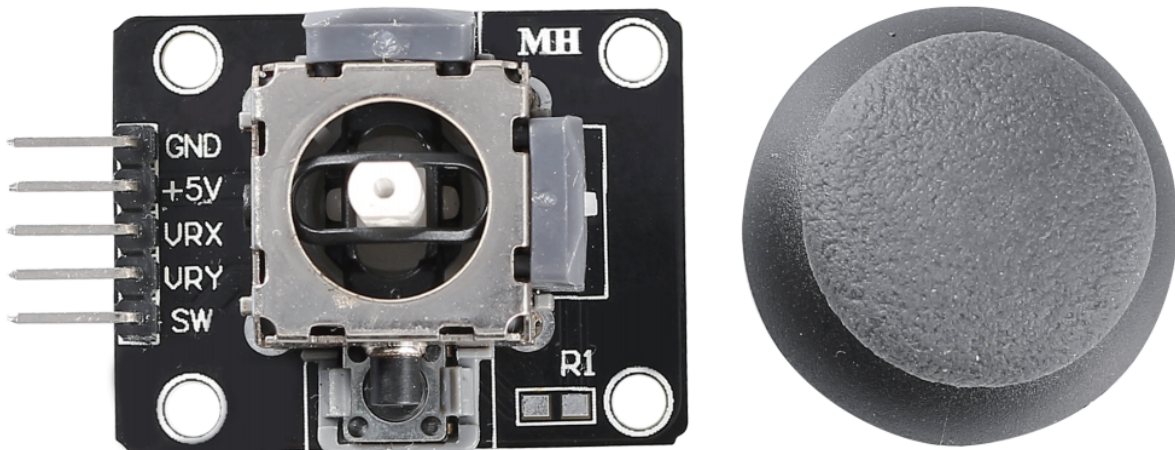
Wenn das Potentiometer als Stromregler wirkt, muss der Schiebekontaktanschluss als einer der Ausgangsanschlüsse verbunden werden.

Wenn Sie mehr über Potentiometer erfahren möchten, siehe: .

Beispiel

- *Potentiometer* (Grundprojekt)
- *HueDial* (Spaßprojekt)

1.2.26 Joystick-Modul

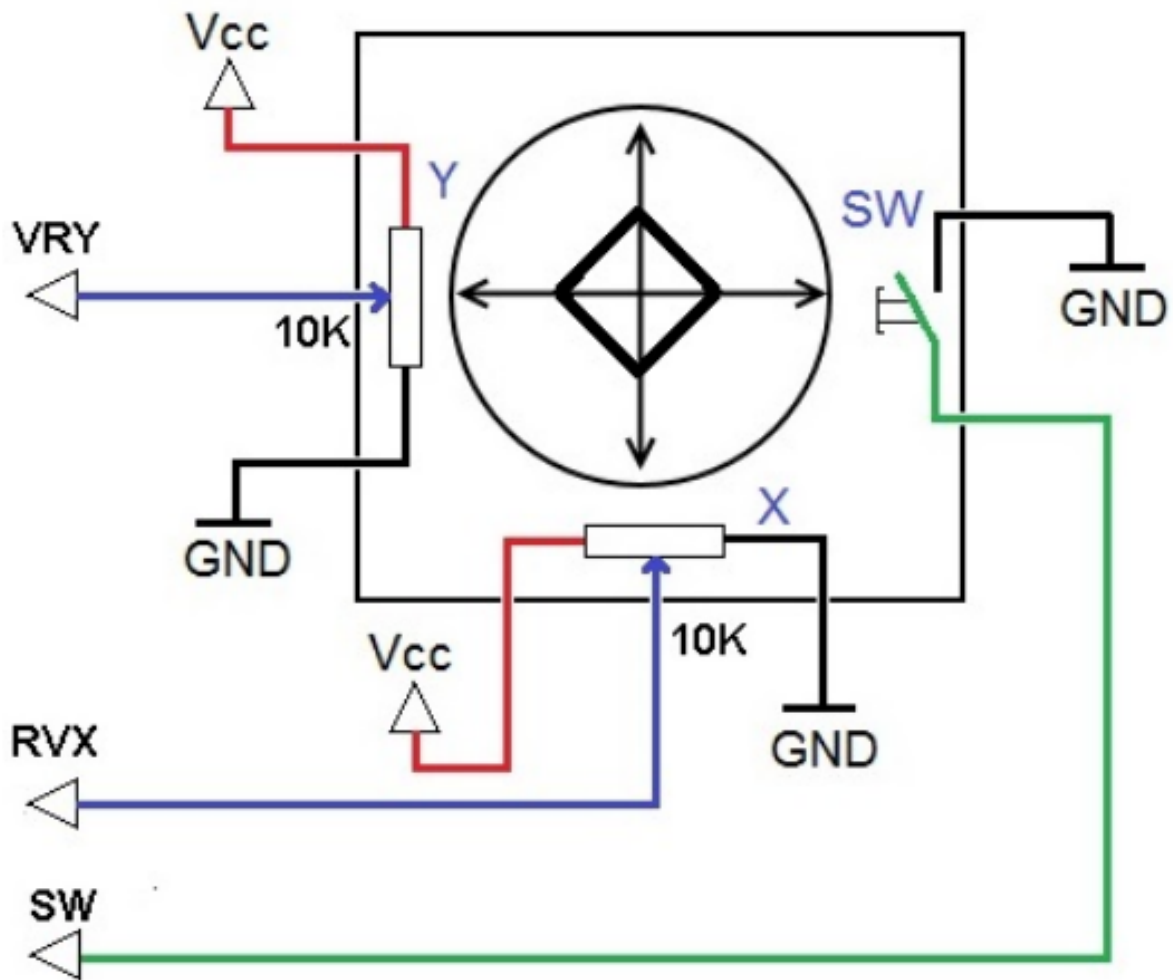


Die grundlegende Idee eines Joysticks besteht darin, die Bewegung eines Sticks in elektronische Informationen umzusetzen, die ein Computer verarbeiten kann.

Um einem Computer eine vollständige Bewegungspalette zu übermitteln, muss ein Joystick die Position des Sticks auf zwei Achsen messen – der X-Achse (links nach rechts) und der Y-Achse (oben und unten). Wie in der Grundgeometrie ermöglichen die X-Y-Koordinaten eine genaue Bestimmung der Position des Sticks.

Um die Position des Sticks zu bestimmen, überwacht das Joystick-Steuerungssystem einfach die Position jeder Welle. Das herkömmliche analoge Joystick-Design macht dies mit zwei Potentiometern oder variablen Widerständen.

Der Joystick verfügt auch über einen digitalen Eingang, der betätigt wird, wenn der Joystick nach unten gedrückt wird.

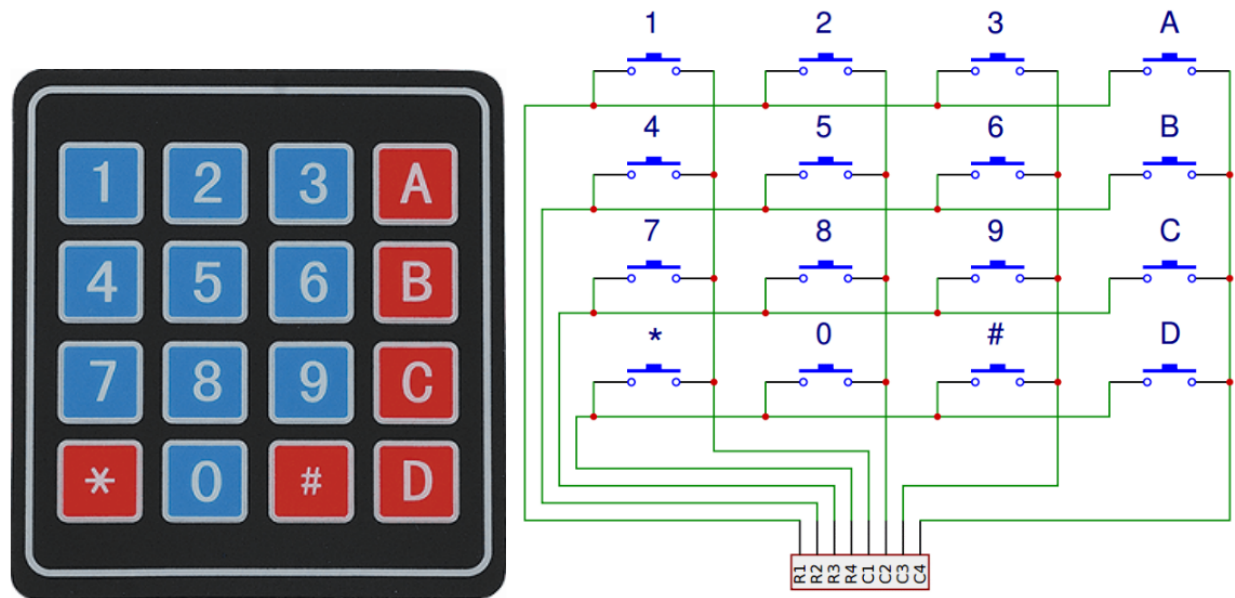


Beispiel

- *Joystick-Modul* (Grundprojekt)
- *SPIEL - Snake* (Spaßprojekt)

1.2.27 Tastenfeld

Ein Tastenfeld ist ein rechteckiges Array aus 12 oder 16 OFF-(ON)-Tasten. Ihre Kontakte sind über einen Header zugänglich, der für die Verbindung mit einem Flachbandkabel oder das Einsetzen in eine gedruckte Schaltungsplatte geeignet ist. Bei einigen Tastenfeldern verbindet jede Taste mit einem separaten Kontakt im Header, während alle Tasten eine gemeinsame Masse teilen.



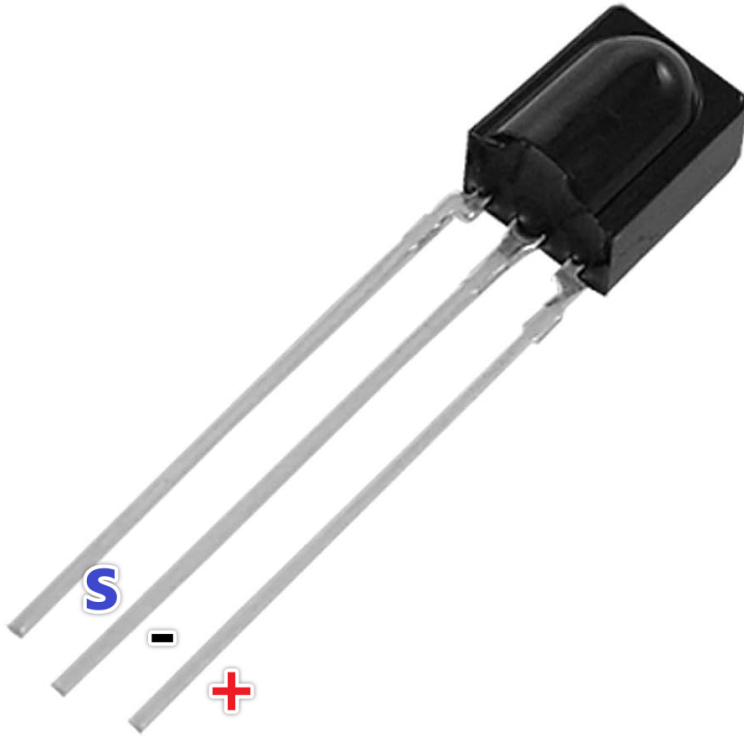
Häufiger sind die Tasten matrixcodiert, was bedeutet, dass jede von ihnen ein einzigartiges Paar von Leitern in einer Matrix überbrückt. Diese Konfiguration eignet sich für das Abfragen durch einen Mikrocontroller, der so programmiert werden kann, dass er nacheinander einen Ausgangsimpuls an jede der vier horizontalen Leitungen sendet. Während jedes Impulses überprüft er die verbleibenden vier vertikalen Leitungen der Reihe nach, um festzustellen, welche davon, wenn überhaupt, ein Signal trägt. Pull-up- oder Pull-down-Widerstände sollten zu den Eingangsleitungen hinzugefügt werden, um ein unvorhersehbares Verhalten der Eingänge des Mikrocontrollers zu verhindern, wenn kein Signal vorhanden ist.

Beispiel

- *Tastefeld* (Grundprojekt)

1.2.28 Infrarotempfänger

IR-Empfänger



- S: Signalausgang
- +: VCC
- -: GND

SL838 ist ein kleiner Empfänger für Infrarot-Fernsteuerungssysteme. Er enthält eine schnelle und hochempfindliche Fotodiode und einen Vorverstärker und ist mit Epoxidharz verpackt, um einen Infrarotfilter zu bilden. Sein Hauptvorteil ist, dass er auch in gestörter Umgebung zuverlässig funktioniert.

Infrarot- oder IR-Kommunikation ist eine beliebte, kostengünstige und einfach zu verwendende drahtlose Kommunikationstechnologie. Infrarotlicht hat eine etwas längere Wellenlänge als sichtbares Licht, daher ist es für das menschliche Auge nicht wahrnehmbar - ideal für drahtlose Kommunikation. Ein gängiges Modulationsschema für die Infrarotkommunikation ist die 38KHz-Modulation.

- Kann für Fernbedienungen verwendet werden
- Breiter Betriebsspannungsbereich: 2,7~5V
- Interner Filter für PCM-Frequenz
- TTL- und CMOS-Kompatibilität
- Starke Störfestigkeit
- RoHS-konform

Fernbedienung



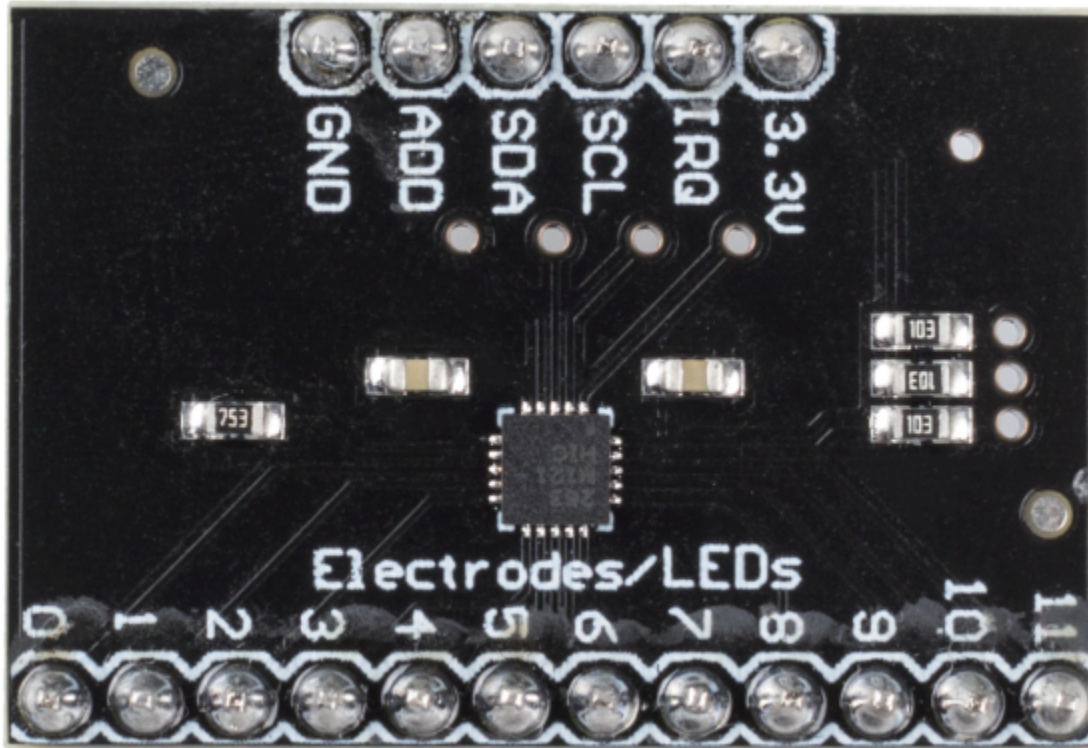
Dies ist eine Mini-Dünn-Infrarot-Fernbedienung mit 21 Funktionstasten und einer Übertragungsdistanz von bis zu 8 Metern, die sich für die Bedienung einer Vielzahl von Geräten in einem Kinderzimmer eignet.

- Größe: 85x39x6mm
- Reichweite der Fernbedienung: 8-10m
- Batterie: 3V Knopfzellen-Lithium-Mangan-Batterie
- Infrarot-Trägerfrequenz: 38KHz
- Oberflächenklebematerial: 0,125mm PET
- Effektive Lebensdauer: mehr als 20.000 Mal

Beispiel

- *Infrarot-Empfänger* (Grundprojekt)
- *SPIEL - Zahl Erraten* (Spaßprojekt)

1.2.29 MPR121



- **3.3V:** Stromversorgung
- **IRQ:** Open-Collector-Interrupt-Ausgangspin, aktiv niedrig
- **SCL:** I2C-Takt
- **SDA:** I2C-Daten
- **ADD:** I2C-Adressauswahl-Eingangspin. Verbindet man den ADDR-Pin mit der VSS-, VDD-, SDA- oder SCL-Leitung, ergeben sich die I2C-Adressen 0x5A, 0x5B, 0x5C und 0x5D
- **GND:** Masse
- **0~11:** Elektrode 0~11, Elektrode ist ein Berührungssensor. Typischerweise können Elektroden einfach ein Stück Metall oder ein Draht sein. Manchmal können jedoch die Länge des Drahtes oder das Material, auf dem die Elektrode liegt, das Auslösen des Sensors erschweren. Aus diesem Grund ermöglicht der MPR121, zu konfigurieren, was erforderlich ist, um eine Elektrode auszulösen und wieder zu deaktivieren.

ÜBERSICHT MPR121

Der MPR121 ist der Nachfolger des kapazitiven Touchsensor-Controllers nach der Erstveröffentlichung der MPR03x-Serie. Der MPR121 verfügt über erhöhte interne Intelligenz, einige der wichtigen Ergänzungen umfassen eine erhöhte Elektrodenanzahl, eine hardwarekonfigurierbare I2C-Adresse, ein erweitertes Filtersystem mit Entprellung und völlig unabhängige Elektroden mit eingebauter Auto-Konfiguration. Das Gerät verfügt auch über einen 13. simulierten Sensor-Kanal, der für die Näherungserkennung mittels multiplexierter Sensoreingänge gewidmet ist.

•

Merkmale

- **Niedriger Stromverbrauch**
 - 1,71 V bis 3,6 V Betriebsspannung

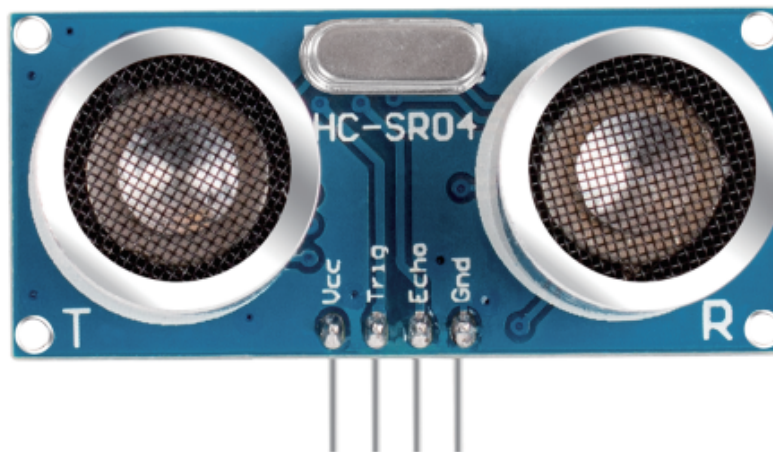
- 29 A Versorgungsstrom bei 16 ms Abtastintervall
- 3 A Strom im Stoppmodus
- **12 Kapazitätssensoreingänge**
 - 8 Eingänge sind multifunktional für LED-Treiber und GPIO
- **Komplette Berührungserkennung**
 - Auto-Konfiguration für jeden Sensoreingang
 - Auto-Kalibrierung für jeden Sensoreingang
 - Berührungs-/Freigabeschwelle und Entprellung für die Berührungserkennung
- I2C-Schnittstelle mit Interrupt-Ausgang
- 3 mm x 3 mm x 0,65 mm 20-poliges QFN-Gehäuse
- Betriebstemperaturbereich von -40°C bis +85°C

Beispiel

- *MPR121* (Grundprojekt)
- *Obstklavier* (Spaßprojekt)

Sensor

1.2.30 Ultraschallmodul



Ein Ultraschallsensor ist ein Gerät zur Messung der Entfernung zu einem Objekt mit Ultraschallwellen. Es besitzt zwei Sonden: eine sendet Ultraschallwellen aus, die andere empfängt diese und wandelt die Übertragungszeit in eine Distanz um. Dies ermöglicht die Ermittlung des Abstands zwischen dem Gerät und einem Hindernis. In der Praxis erweist es sich als sehr praktisch und funktionell.

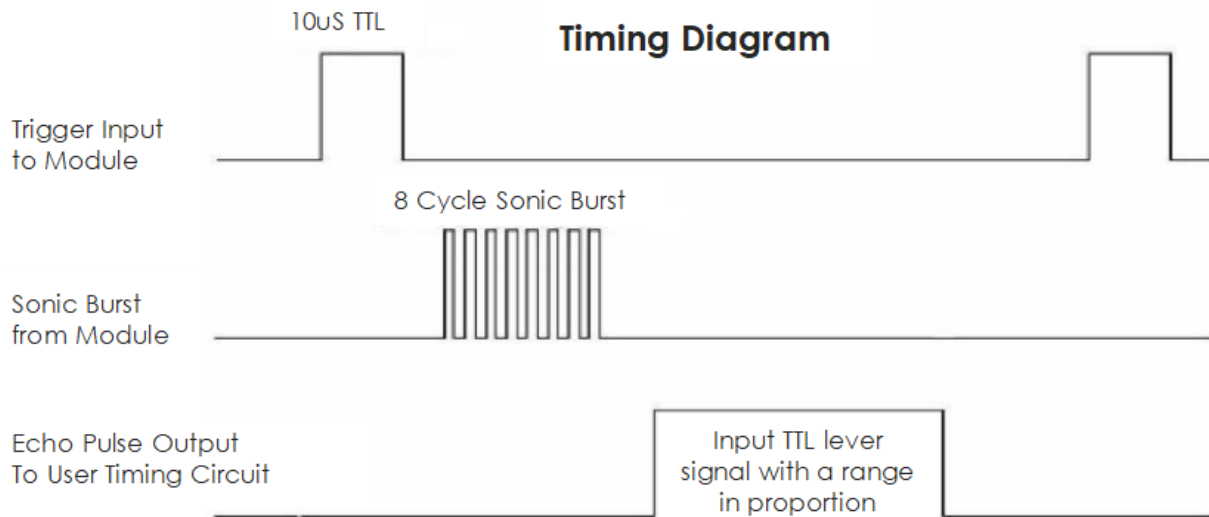
Das Modul bietet eine berührungslose Messfunktion von 2 cm bis 400 cm, mit einer Genauigkeit von bis zu 3 mm. Es gewährleistet eine stabile Signalübertragung bis zu einer Entfernung von 5 m; darüber hinaus nimmt die Signalstärke ab und verschwindet bei etwa 7 m.

Das Modul umfasst Ultraschallsender, -empfänger und Steuerschaltkreise. Die grundlegenden Prinzipien sind wie folgt:

1. Ein IO-Flip-Flop verarbeitet ein Hochpegelsignal von mindestens 10µs.
2. Das Modul sendet automatisch acht 40kHz-Signale und prüft, ob ein Pulssignal zurückkehrt.

3. Wenn das Signal zurückkehrt, wird die Dauer des Hochpegelausgangs des IOs zur Zeit vom Aussenden bis zum Zurückkehren der Ultraschallwelle. Hier gilt: Testdistanz = (Hochzeit x Schallgeschwindigkeit (340 m/s) / 2.

Das Timing-Diagramm ist unten dargestellt.



Sie müssen nur einen kurzen 10µs-Impuls an den Trigger-Eingang senden, um die Entfernungsmessung zu starten. Das Modul sendet dann einen 8-Zyklus-Burst von Ultraschall mit 40 kHz aus und erhöht sein Echo. Die Entfernung kann durch das Zeitintervall zwischen dem Senden des Trigger-Signals und dem Empfangen des Echosignals berechnet werden.

Formel: $\text{us} / 58 = \text{Zentimeter}$ oder $\text{us} / 148 = \text{Zoll}$; oder: die Entfernung = Hochpegelzeit * Geschwindigkeit (340M/S) / 2; es wird empfohlen, einen Messzyklus von über 60ms zu verwenden, um Signalüberschneidungen zwischen Trigger- und Echosignal zu vermeiden.

Beispiel

- *Ultraschall* (Grundprojekt)
- *Smarter Mülleimer* (Spaßprojekt)

1.2.31 Bodenfeuchtigkeitsmodul

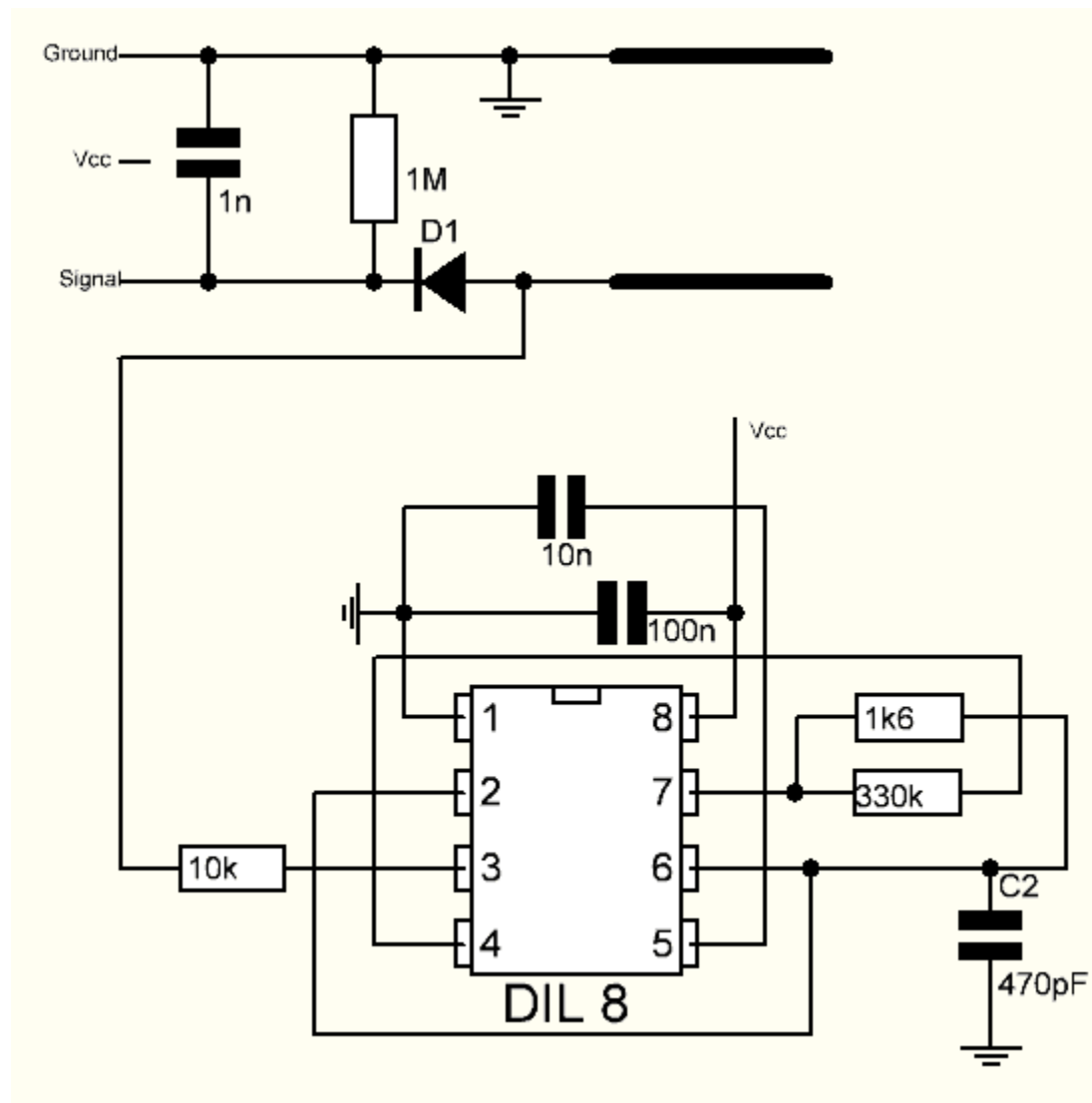


- GND: Erde
- VCC: Stromversorgung, 3.3v~5V
- AUOT: Gibt den Wert der Bodenfeuchtigkeit aus, je feuchter der Boden, desto geringer sein Wert.

Dieser kapazitive Bodenfeuchtigkeitssensor unterscheidet sich von den meisten auf dem Markt erhältlichen resistiven Sensoren, da er das Prinzip der kapazitiven Induktion zur Messung der Bodenfeuchtigkeit nutzt. Dies vermeidet das Problem der Korrosionsanfälligkeit bei resistiven Sensoren und verlängert dadurch die Lebensdauer erheblich.

Er besteht aus korrosionsbeständigen Materialien und hat eine ausgezeichnete Lebensdauer. Setzen Sie ihn in die Erde um Pflanzen herum ein, um Echtzeitdaten zur Bodenfeuchtigkeit zu überwachen. Das Modul beinhaltet einen Spannungsregler, der es ermöglicht, in einem Spannungsbereich von 3,3 ~ 5,5 V zu arbeiten. Es eignet sich ideal für Niederspannungs-Mikrocontroller mit 3,3 V und 5 V Versorgung.

Das Hardware-Schaltbild des kapazitiven Bodenfeuchtigkeitssensors ist unten dargestellt.



Es gibt einen Festfrequenz-Oszillator, der mit einem 555-Timer-IC aufgebaut ist. Die erzeugte Rechteckwelle wird dann wie ein Kondensator an den Sensor geführt. Allerdings hat der Kondensator für das Rechteckwellensignal eine gewisse Reaktanz oder, um es anders auszudrücken, einen Widerstand mit einem rein ohmschen Widerstand (10k-Widerstand an Pin 3), um einen Spannungsteiler zu bilden.

Je höher die Bodenfeuchtigkeit, desto höher die Kapazität des Sensors. Infolgedessen hat die Rechteckwelle weniger Reaktanz, was die Spannung auf der Signalleitung verringert, und der Wert des analogen Eingangs über den Mikrocontroller ist kleiner.

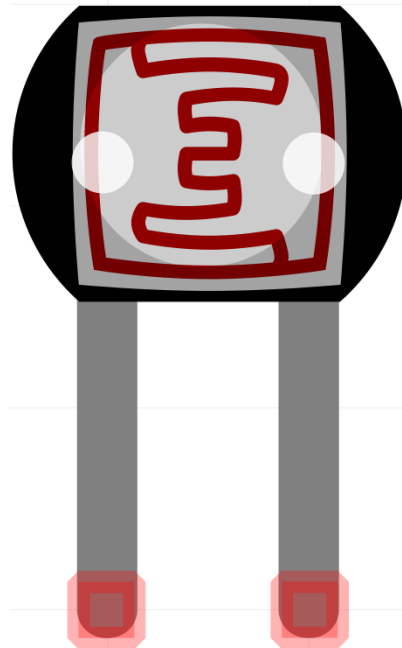
Spezifikation

- Betriebsspannung: 3,3 ~ 5,5 VDC
- Ausgangsspannung: 0 ~ 3,0VDC
- Betriebsstrom: 5mA
- Schnittstelle: PH2.0-3P
- Abmessungen: 3,86 x 0,905 Zoll (L x B)
- Gewicht: 15g

Beispiel

- *Bodenfeuchtigkeit* (Grundprojekt)
- *Pflanzenmonitor* (Spaßprojekt)

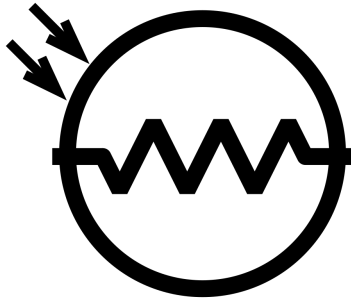
1.2.32 Fotowiderstand



Ein Fotowiderstand, auch Fotoleiter genannt, ist ein lichtgesteuerter variabler Widerstand. Der Widerstand eines Fotowiderstands nimmt mit steigender Lichtintensität ab; anders ausgedrückt, zeigt er eine Fotoleitfähigkeit.

Fotowiderstände können in lichtempfindlichen Detektorschaltungen sowie in licht- und dunkelaktivierten Schaltkreisen als Widerstands-Halbleiter eingesetzt werden. Im Dunkeln kann der Widerstand eines Fotowiderstands mehrere Megaohm (M) betragen, während er im Hellen auf wenige hundert Ohm sinken kann.

Hier ist das elektronische Symbol eines Fotowiderstands.

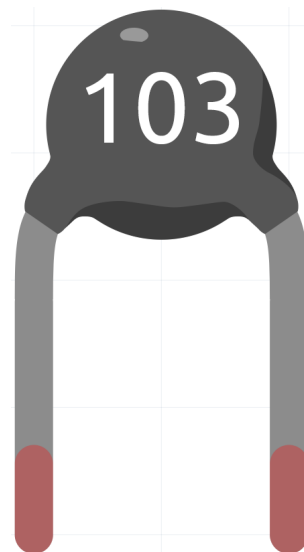


-

Beispiel

- *Fotowiderstand* (Grundprojekt)
- *Lichtempfindliches Array* (Spaßprojekt)

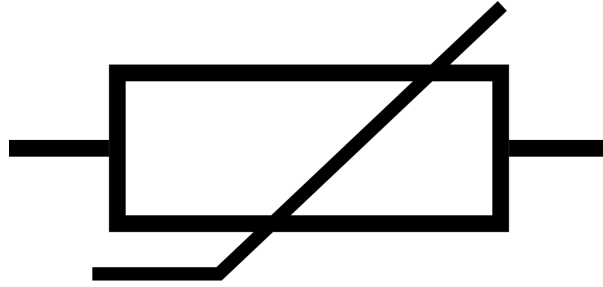
1.2.33 Thermistor



Ein Thermistor ist eine Art Widerstand, dessen Widerstand stark temperaturabhängig ist, stärker als bei herkömmlichen Widerständen. Das Wort ist eine Kombination aus „thermal“ (thermisch) und „resistor“ (Widerstand). Thermistoren werden weit verbreitet eingesetzt, unter anderem als Einschaltstrombegrenzer, Temperatursensoren (typischerweise vom Typ NTC, also mit negativem Temperaturkoeffizienten), selbstzurücksetzende Überstromschutzgeräte und selbst-regulierende Heizelemente (typischerweise vom Typ PTC, also mit positivem Temperaturkoeffizienten).

-

Hier ist das elektronische Symbol eines Thermistors.



Thermistoren gibt es in zwei grundlegend entgegengesetzten Typen:

- Bei NTC-Thermistoren sinkt der Widerstand mit steigender Temperatur, in der Regel aufgrund einer Zunahme an Leitungselektronen, die durch thermische Agitation aus dem Valenzband gehoben werden. Ein NTC wird häufig als Temperatursensor eingesetzt oder in Reihe mit einem Schaltkreis als Einschaltstrombegrenzer verwendet.
- Bei PTC-Thermistoren steigt der Widerstand mit zunehmender Temperatur, in der Regel aufgrund erhöhter thermischer Gitterschwingungen, insbesondere jener von Verunreinigungen und Fehlstellen. PTC-Thermistoren werden üblicherweise in Reihe mit einem Schaltkreis installiert und dienen als Schutz gegen Überstrombedingungen, als rückstellbare Sicherungen.

In diesem Kit verwenden wir einen NTC-Typ. Jeder Thermistor hat einen normalen Widerstand. Hier beträgt er 10k Ohm, gemessen bei 25 Grad Celsius.

Hier ist die Beziehung zwischen Widerstand und Temperatur:

$$R_T = R_N * \exp(B(1/T_K - 1/T_N))$$

- **R_T** ist der Widerstand des NTC-Thermistors bei der Temperatur T_K .
- **R_N** ist der Widerstand des NTC-Thermistors unter der Nenntemperatur T_N . Hier beträgt der numerische Wert von R_N 10k.
- **T_K** ist eine Kelvin-Temperatur, und die Einheit ist K. Hier beträgt der numerische Wert von T_K 273,15 + Grad Celsius.
- **T_N** ist eine Nenn-Kelvin-Temperatur; die Einheit ist ebenfalls K. Hier beträgt der numerische Wert von T_N 273,15+25.
- Und **$B(\text{beta})$** , die Materialkonstante des NTC-Thermistors, wird auch als Wärmeempfindlichkeitsindex bezeichnet und hat den numerischen Wert 3950.
- **\exp** ist die Abkürzung für Exponentialfunktion, und die Basiszahl e ist eine natürliche Zahl und ungefähr gleich 2,7.

Konvertieren Sie diese Formel $T_K = 1 / (\ln(R_T/R_N)/B + 1/T_N)$, um die Kelvin-Temperatur zu erhalten, von der 273,15 abgezogen die Grad Celsius ergibt.

Diese Beziehung ist eine empirische Formel. Sie ist nur genau, wenn Temperatur und Widerstand innerhalb des wirkamen Bereichs liegen.

Beispiel

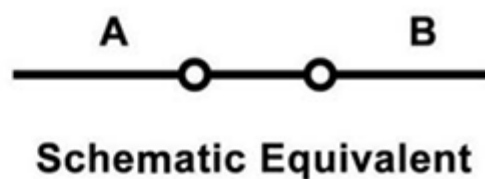
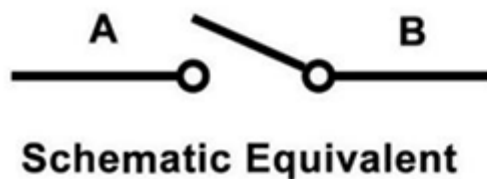
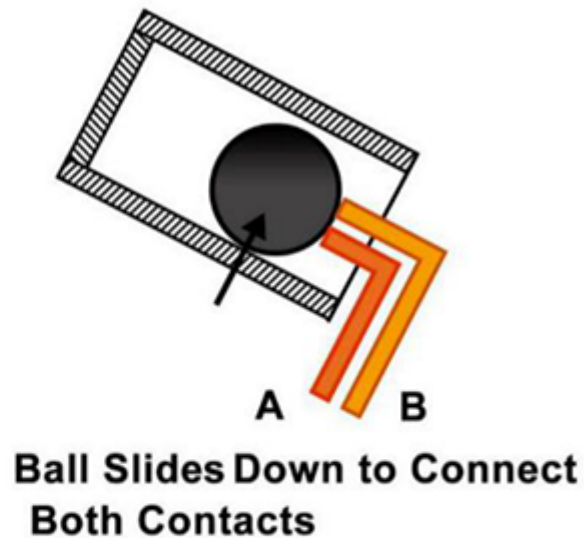
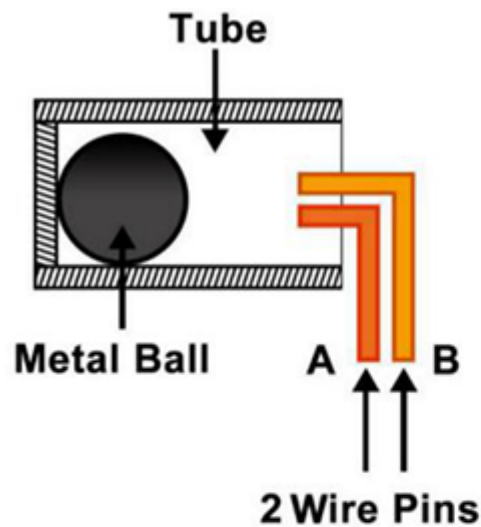
- *Thermistor* (Grundprojekt)
- *Smarter Ventilator* (Spaßprojekt)

1.2.34 Neigungsschalter



Der hier verwendete Neigungsschalter ist ein Kugeltyp mit einer Metallkugel im Inneren. Er dient zur Erkennung von Neigungen in kleinen Winkeln.

Das Prinzip ist sehr einfach. Wenn der Schalter in einem bestimmten Winkel geneigt wird, rollt die Kugel nach unten und berührt die beiden Kontakte, die mit den äußeren Pins verbunden sind, und löst so Schaltkreise aus. Andernfalls bleibt die Kugel von den Kontakten entfernt und unterbricht dadurch die Schaltkreise.



Beispiel

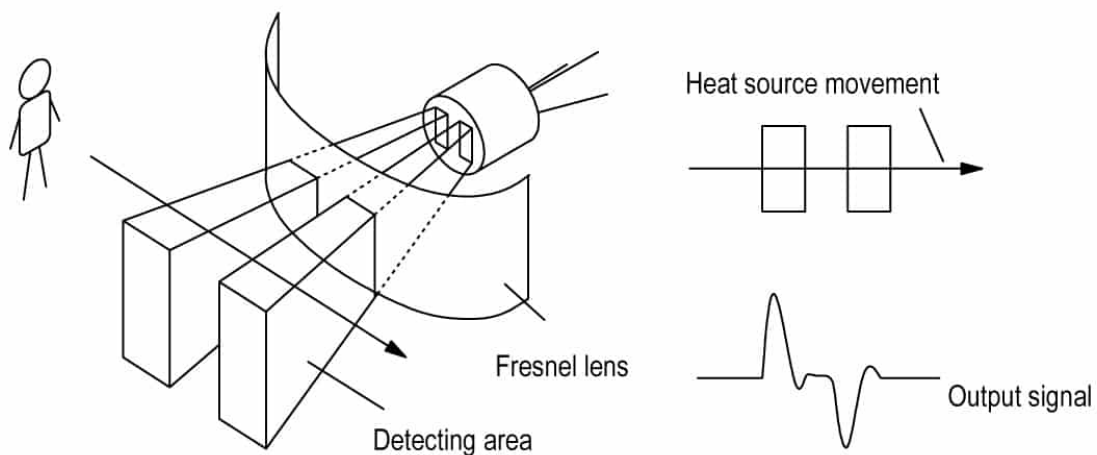
- *Kippschalter* (Grundprojekt)

1.2.35 PIR-Bewegungssensormodul

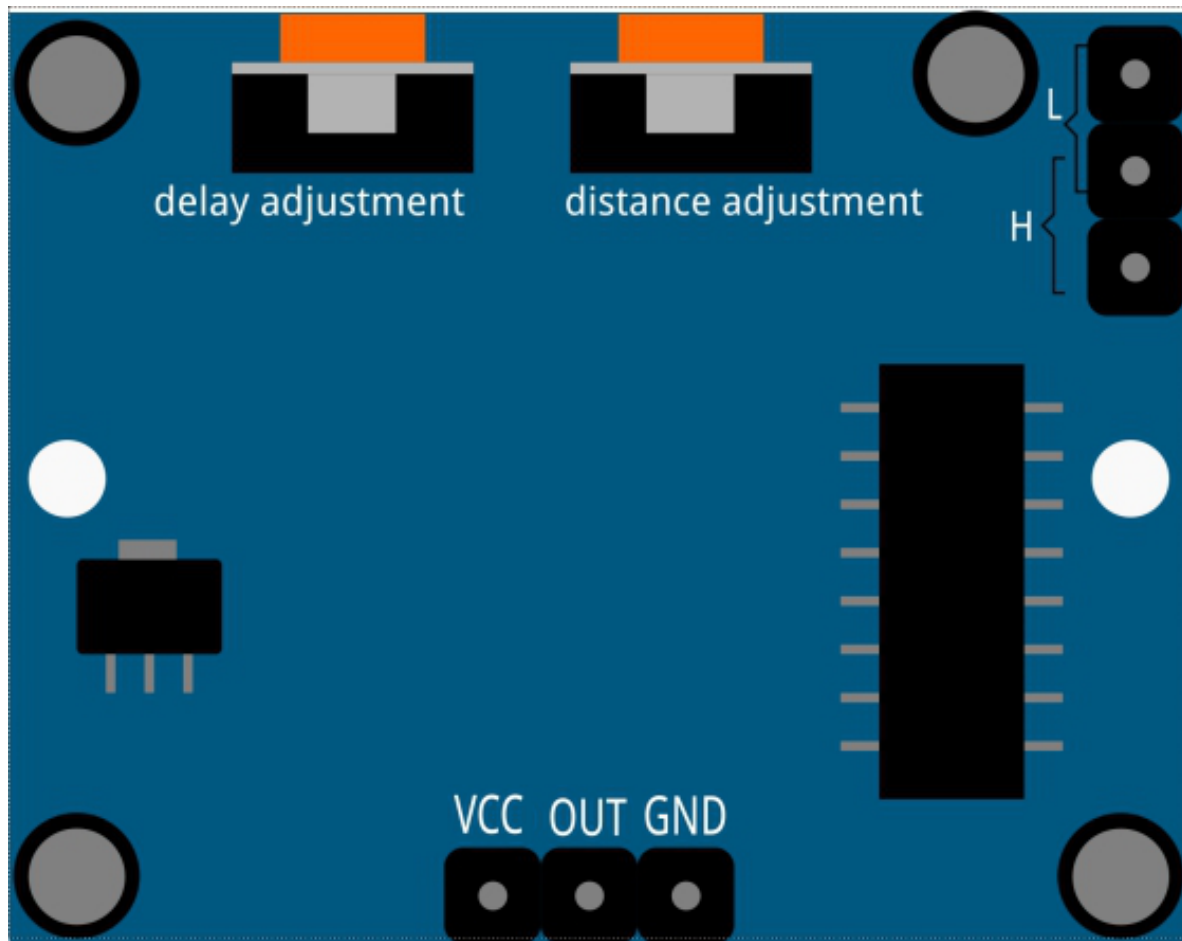


Der PIR-Sensor erkennt infrarote Wärmestrahlung, die zur Erkennung der Anwesenheit von Organismen verwendet werden kann, die infrarote Wärme abstrahlen.

Der PIR-Sensor ist in zwei Schlitze unterteilt, die mit einem Differenzverstärker verbunden sind. Solange ein stationäres Objekt vor dem Sensor ist, erhalten die beiden Schlitze dieselbe Menge an Strahlung und der Ausgangswert ist null. Bewegt sich jedoch ein Objekt vor dem Sensor, empfängt einer der Schlitze mehr Strahlung als der andere, wodurch der Ausgang schwankt. Diese Änderung der Ausgangsspannung ist ein Ergebnis der Bewegungserkennung.



Nachdem das Sensormodul verkabelt ist, erfolgt eine einminütige Initialisierung. Während der Initialisierung gibt das Modul 0-3 Mal in Intervallen ein Signal aus. Dann geht das Modul in den Standby-Modus. Bitte vermeiden Sie die Interferenz von Lichtquellen und anderen Quellen an der Oberfläche des Moduls, um Fehlfunktionen durch störende Signale zu vermeiden. Es ist sogar besser, das Modul ohne zu viel Wind zu verwenden, da der Wind ebenfalls den Sensor stören kann.



Einstellung der Distanz

Durch Drehen des Knopfes des Distanzeinstellungspotentiometers im Uhrzeigersinn erhöht sich die Reichweite der Erfassungsdistanz, und die maximale Erfassungsdistanz beträgt etwa 0-7 Meter. Wenn man es gegen den Uhrzeigersinn dreht, verringert sich die Reichweite der Erfassungsdistanz, und die minimale Erfassungsdistanz beträgt etwa 0-3 Meter.

Verzögerungseinstellung

Drehen Sie den Knopf des Verzögerungseinstellungspotentiometers im Uhrzeigersinn, um die Erfassungsverzögerung zu erhöhen. Die maximale Erfassungsverzögerung kann bis zu 300s betragen. Im Gegenteil, wenn Sie es gegen den Uhrzeigersinn drehen, können Sie die Verzögerung verkürzen, mit einem Minimum von 5s.

Zwei Auslösemodi

Wählen Sie verschiedene Modi mit der Jumperkappe.

- **H:** Wiederholbarer Auslösemodus, nachdem der menschliche Körper erkannt wurde, gibt das Modul ein hohes Signal aus. Während der nachfolgenden Verzögerungsperiode, wenn jemand den Erfassungsbereich betritt, bleibt das Ausgangssignal hoch.
- **L:** Nicht-wiederholbarer Auslösemodus, gibt ein hohes Signal aus, wenn es den menschlichen Körper erfasst. Nach der Verzögerung wechselt der Ausgang automatisch von hoch auf niedrig.

Beispiele

- *PIR-Bewegungssensormodul* (Grundprojekt)
- *Willkommen* (Spaßprojekt)

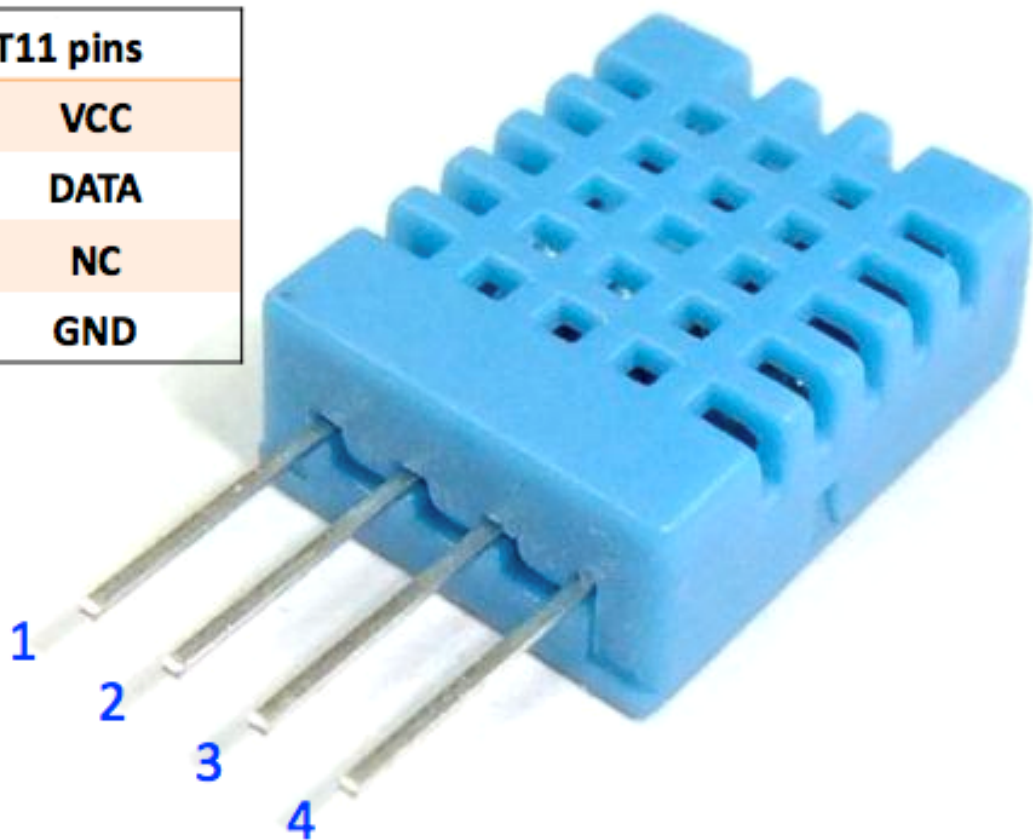
- *Sicherheitssystem über IFTTT* (IoT-Projekt)

1.2.36 Feuchtigkeitssensor-Modul

Der digitale Temperatur- und Feuchtigkeitssensor DHT11 ist ein Verbundsensor, der einen kalibrierten digitalen Signalausgang für Temperatur und Feuchtigkeit enthält. Die Technologie von speziellen digitalen Modulsammlungen und die Temperatur- und Feuchtigkeitserfassungstechnologie werden angewandt, um sicherzustellen, dass das Produkt eine hohe Zuverlässigkeit und ausgezeichnete Langzeitstabilität aufweist.

Nur drei Pins stehen zur Verwendung bereit: VCC, GND und DATA. Der Kommunikationsprozess beginnt damit, dass die DATA-Leitung Startsignale an DHT11 sendet, und DHT11 empfängt die Signale und gibt ein Antwortsignal zurück. Dann empfängt der Host das Antwortsignal und beginnt, 40-Bit-Feuchtigkeitsdaten zu empfangen (8-Bit-Feuchtigkeit ganz + 8-Bit-Feuchtigkeit Dezimal + 8-Bit-Temperatur ganz + 8-Bit-Temperatur Dezimal + 8-Bit-Prüfsumme).

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



•

Beispiel

- *Feuchtigkeitssensor-Modul* (Grundprojekt)
- *Pflanzenmonitor* (Spaßprojekt)
- *Arduino IoT Cloud* (IoT-Projekt)
- *Bluetooth-Umweltmonitor* (IoT-Projekt)

1.2.37 MFRC522-Modul

RFID

Radiofrequenzidentifikation (RFID) bezieht sich auf Technologien, die drahtlose Kommunikation zwischen einem Objekt (oder Tag) und einem abfragenden Gerät (oder Lesegerät) zur automatischen Verfolgung und Identifizierung dieser Objekte verwenden. Die Übertragsreichweite des Tags ist auf einige Meter vom Lesegerät begrenzt. Eine direkte Sichtlinie zwischen Lesegerät und Tag ist nicht unbedingt erforderlich.

Die meisten Tags enthalten mindestens einen integrierten Schaltkreis (IC) und eine Antenne. Der Mikrochip speichert Informationen und ist verantwortlich für das Management der Funkfrequenzkommunikation mit dem Lesegerät. Passive Tags haben keine unabhängige Energiequelle und hängen von einem externen elektromagnetischen Signal, das vom Lesegerät bereitgestellt wird, für ihren Betrieb ab. Aktive Tags verfügen über eine unabhängige Energiequelle, wie eine Batterie, und können daher über erhöhte Verarbeitungs-, Übertragungsfähigkeiten und Reichweite verfügen.



MFRC522

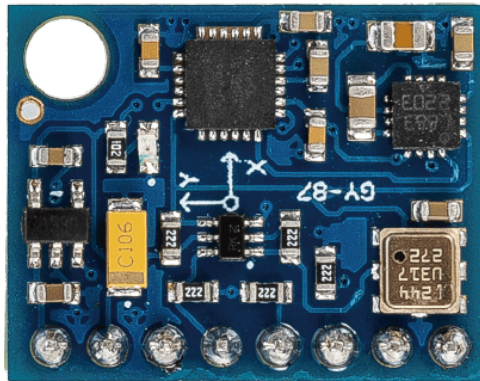
Der MFRC522 ist eine Art integrierter Lese- und Schreibchip für Karten. Er wird üblicherweise im Funkbereich bei 13,56 MHz eingesetzt. Von der NXP Company eingeführt, ist er ein Niederspannungs-, kostengünstiger und kleinformatiger kontaktloser Kartenchip, eine ideale Wahl für intelligente Instrumente und tragbare Handgeräte.

Der MF RC522 verwendet fortschrittliche Modulations- und Demodulationskonzepte, die vollständig in allen Arten von 13,56 MHz passiven kontaktlosen Kommunikationsmethoden und -protokollen präsentiert werden. Zusätzlich unterstützt er den schnellen CRYPTO1-Verschlüsselungsalgorithmus zur Verifizierung von MIFARE-Produkten. Der MFRC522 unterstützt auch die MIFARE-Serie von hochgeschwindigkeitskontaktloser Kommunikation mit einer bidirektionalen Datenübertragungsrate von bis zu 424 kbit/s. Als neues Mitglied der 13,56 MHz hochintegrierten Leserkartenserie ist der MF RC522 dem vorhandenen MF RC500 und MF RC530 ähnlich, aber es gibt auch große Unterschiede. Er kommuniziert mit der Hostmaschine über die serielle Art, die weniger Verkabelung benötigt. Sie können zwischen SPI-, I2C- und seriellen UART-Modus (ähnlich wie RS232) wählen, was hilft, die Verbindung zu reduzieren, Platz auf der PCB-Platine zu sparen (kleinere Größe) und die Kosten zu senken.

Beispiele

- *RFID-RC522 Modul* (Grundprojekt)
- *Zugangskontrollsystem* (Spaßprojekt)

1.2.38 GY-87 IMU-Modul



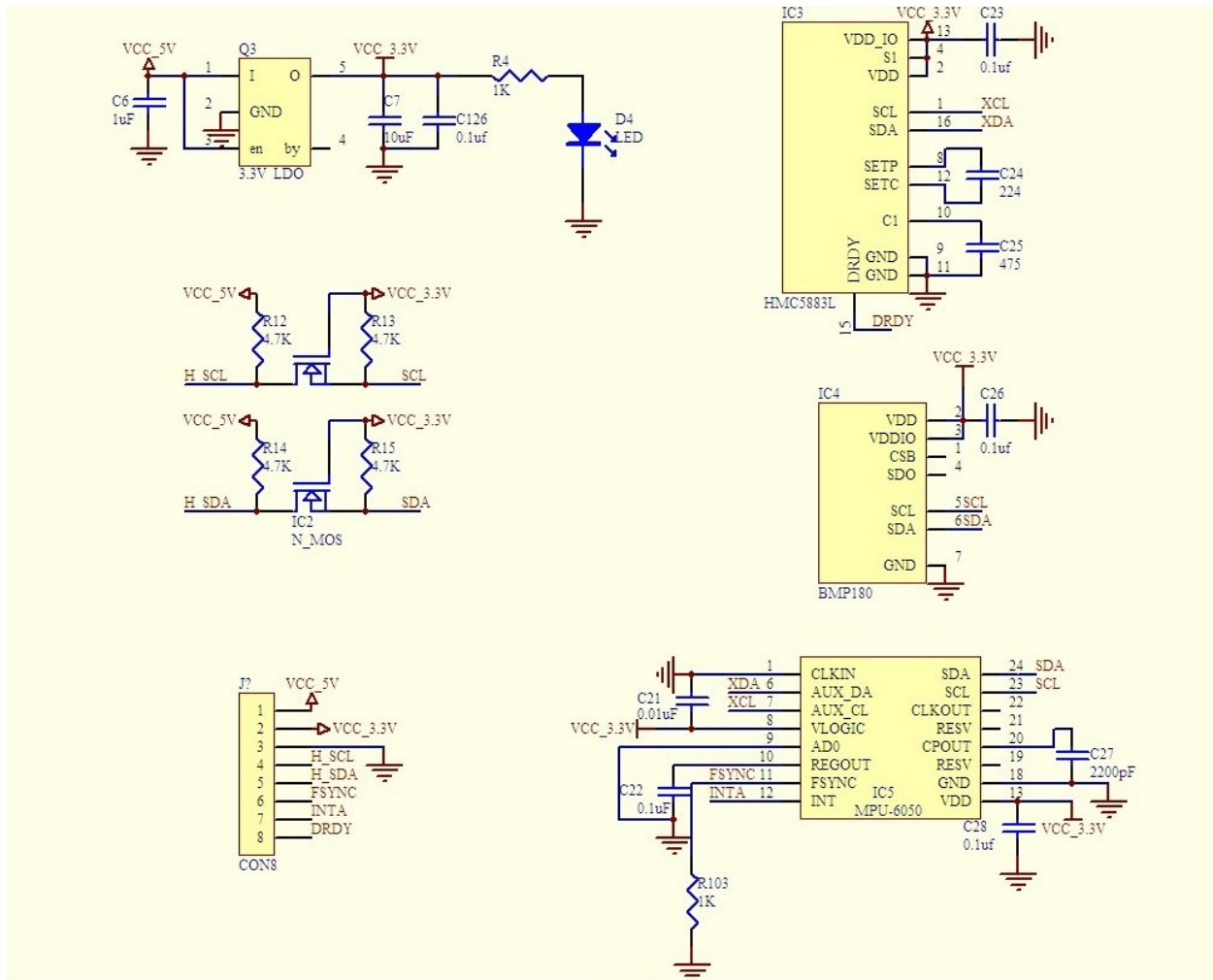
Das GY-87-Sensormodul ist ein hochpräzises 10-Achsen-Modul (10DOF), das in der Lage ist, Beschleunigung, Winkelgeschwindigkeit und magnetische Feldstärke entlang drei Achsen: x, y und z zu messen. Es besteht aus drei Hauptkomponenten: MPU6050, QMC5883L und BMP180 und kommuniziert über das I2C-Protokoll.

Das GY-87-Sensormodul basiert auf drei Sensoren:

1. **MPU6050:** Dies ist ein 6-Achsen-Beschleunigungsmesser und Gyroskop, das Beschleunigung und Winkelgeschwindigkeit in den drei Achsen x, y und z messen kann.
2. **QMC5883L:** Dies ist ein 3-Achsen-Digitalkompass, der die Stärke des Magnetfelds in den drei Achsen x, y und z messen kann.
3. **BMP180:** Dies ist ein barometrischer Temperatur- und Drucksensor, der den atmosphärischen Druck und die Temperatur messen kann.

Der MPU6050 misst Beschleunigung und Winkelgeschwindigkeit in den drei Achsen x, y und z. Der QMC5883L misst die Stärke des Magnetfelds in den drei Achsen x, y und z. Der BMP180 misst den atmosphärischen Druck und die Temperatur. Die Daten dieser Sensoren werden kombiniert, um genaue Informationen über die Orientierung des Moduls im Raum zu liefern.

Das GY-87-Sensormodul wird häufig in Anwendungen wie Drohnen, Robotik und anderen Projekten verwendet, die genaue Orientierungsinformationen benötigen. Es ist mit Arduino-Boards kompatibel und kann leicht über das I2C-Kommunikationsprotokoll mit ihnen verbunden werden.



Beispiel

- *BMP180* (Grundprojekt)
- *MPU6050* (Grundprojekt)
- *QMC5883L* (Grundprojekt)
- *SPIEL - Flucht* (Spaßprojekt)

Einstieg in Arduino

Wenn Sie keine Ahnung von Arduino haben. Hier sind einige Begriffe, die ich Ihnen zeigen möchte: Elektronik, Design, Programmierung und sogar Maker. Einige von Ihnen mögen denken, dass diese Worte weit entfernt von uns sind, aber in Wirklichkeit sind sie es überhaupt nicht. Denn Arduino kann uns in die Welt der Programmierung führen und uns helfen, den Traum, ein Maker zu sein, zu verwirklichen. In dieser Sitzung werden wir lernen:

- Was ist Arduino?
- Was kann Arduino tun?
- Wie baut man ein Arduino-Projekt?

2.1 Was ist Arduino?

Zunächst möchte ich Ihnen eine kurze Einführung in Arduino geben.

Arduino ist eine praktische, flexible und benutzerfreundliche Open-Source-Plattform für elektronisches Prototyping, die sowohl Hardware-Arduino-Boards verschiedener Modelle als auch die Software Arduino IDE umfasst. Sie eignet sich nicht nur für Ingenieure zum schnellen Prototyping, sondern auch für Künstler, Designer, Hobbyisten und ist fast ein unverzichtbares Werkzeug für moderne Maker.

Arduino ist ein ziemlich umfassendes System. Es umfasst Software, Hardware und eine riesige Online-Community von Menschen, die sich nie getroffen haben, aber dank eines gemeinsamen Hobbys zusammenarbeiten können. Jeder in der Arduino-Familie nutzt seine Weisheit, schafft mit seinen Händen und teilt eine großartige Erfindung nach der anderen. Und auch Sie können ein Teil davon sein.

2.2 Was kann Arduino tun?

Vielleicht fragen Sie sich, was Arduino eigentlich kann. Kurz gesagt, Arduino wird all Ihre Probleme lösen.

Technisch gesehen ist Arduino ein programmierbarer Logikcontroller. Es handelt sich um ein Entwicklungsboard, mit dem viele spannende und kreative elektronische Kreationen erstellt werden können: wie ferngesteuerte Autos, Roboterarme, bionische Roboter, Smart Homes usw.

Arduino-Boards sind unkompliziert, einfach und leistungsfähig, geeignet für Schüler, Maker und sogar professionelle Programmierer.

Bis heute entwickeln Elektronik-Enthusiasten weltweit auf Basis von Arduino-Entwicklungsboards kreative elektronische Kreationen.

2.3 Wie man ein Arduino-Projekt aufbaut

Folgen Sie diesen Schritten, um zu lernen, wie Sie Arduino von Grund auf nutzen!

2.3.1 Download und Installation der Arduino IDE 2.0

Die Arduino IDE, bekannt als Arduino Integrated Development Environment, bietet alle notwendigen Softwareunterstützungen, um ein Arduino-Projekt zu vollenden. Es handelt sich um eine speziell für Arduino entwickelte Programmiersoftware, bereitgestellt vom Arduino-Team, die es uns ermöglicht, Programme zu schreiben und auf das Arduino-Board hochzuladen.

Die Arduino IDE 2.0 ist ein Open-Source-Projekt. Sie stellt einen großen Schritt gegenüber ihrem robusten Vorgänger, der Arduino IDE 1.x, dar und kommt mit einer überarbeiteten Benutzeroberfläche, verbessertem Board- & Bibliotheksmanager, Debugger, Autocomplete-Funktion und vielem mehr.

In diesem Tutorial zeigen wir, wie man die Arduino IDE 2.0 auf Ihrem Windows-, Mac- oder Linux-Computer herunterlädt und installiert.

Voraussetzungen

- Windows - Win 10 und neuer, 64 Bit
- Linux - 64 Bit
- Mac OS Intel - Version 10.14: „Mojave“ oder neuer, 64 Bit
- Mac OS Apple Silicon - Version 11: „Big Sur“ oder neuer, 64 Bit

Download der Arduino IDE 2.0

1. Besuchen Sie .
2. Laden Sie die IDE für Ihre Betriebssystemversion herunter.



Arduino IDE 2.0.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

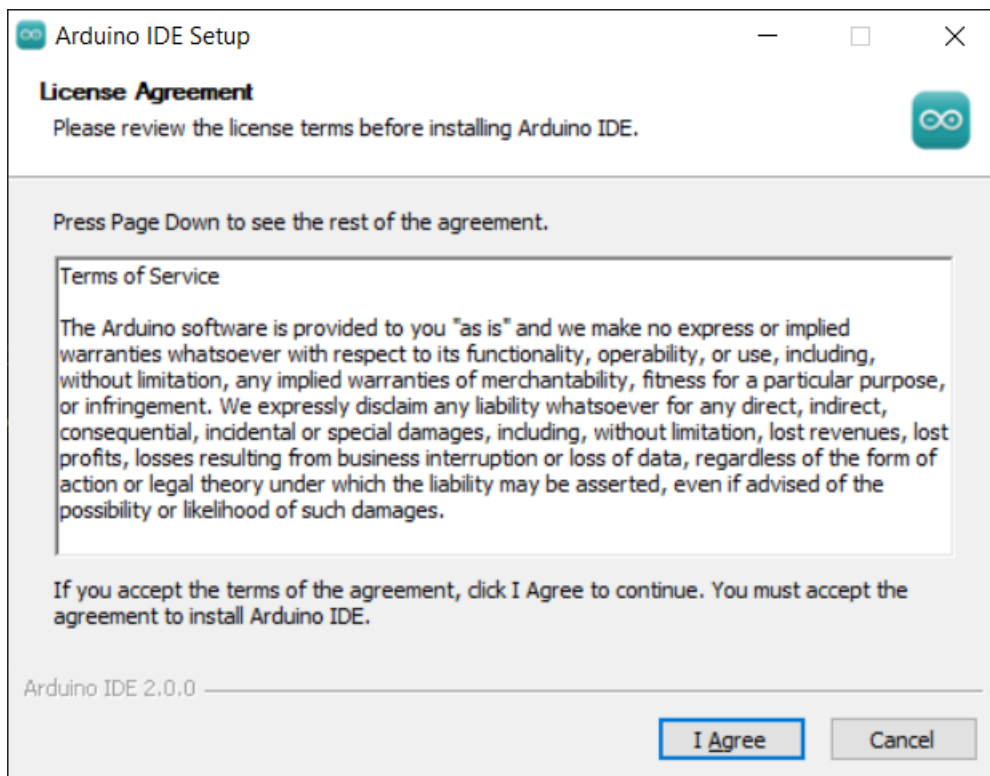
Linux ZIP file 64 bits (X86-64)

macOS 10.14: "Mojave" or newer, 64 bits

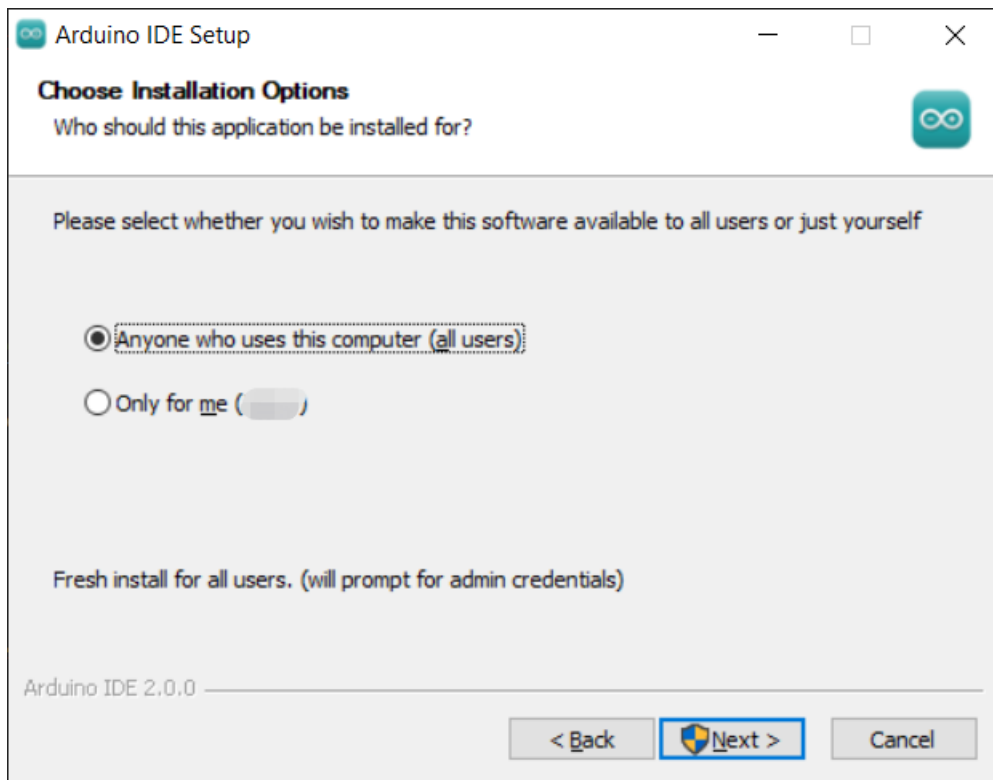
Installation

Windows

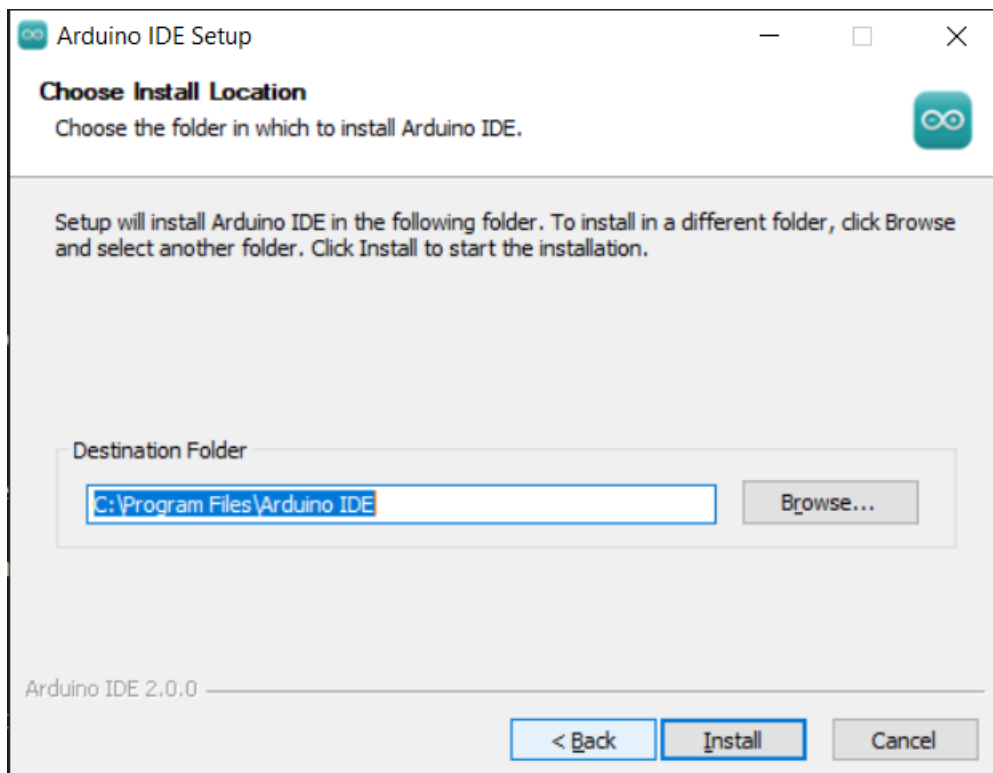
1. Doppelklicken Sie auf die Datei `arduino-ide_xxxx.exe`, um die heruntergeladene Datei auszuführen.
2. Lesen Sie die Lizenzvereinbarung und stimmen Sie ihr zu.



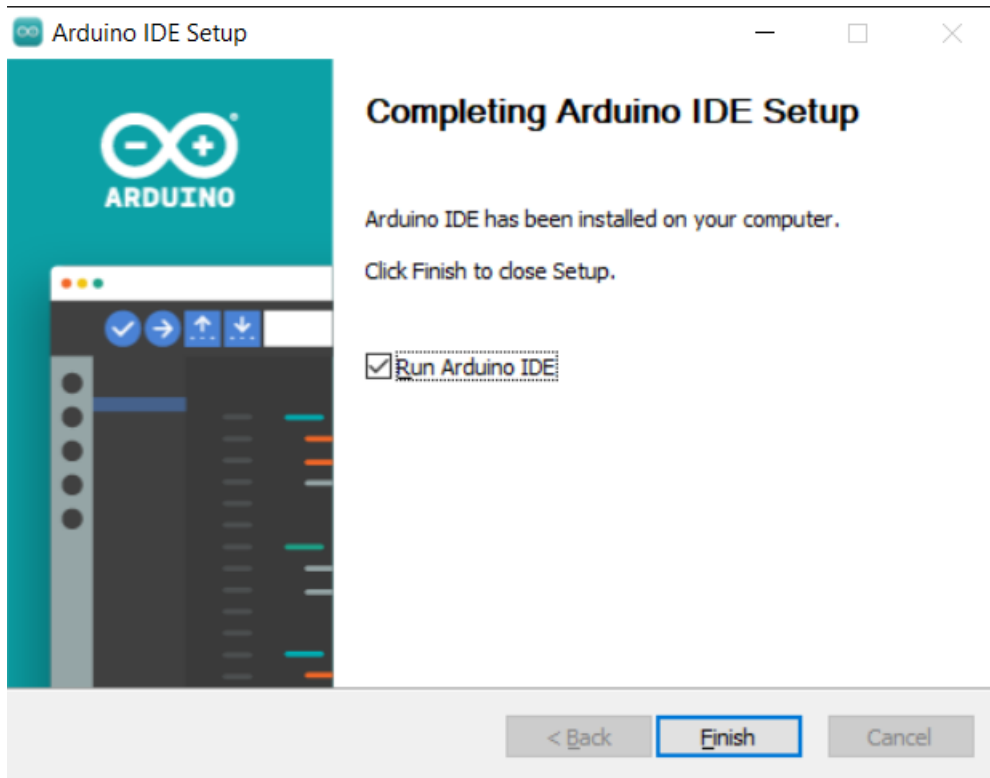
3. Wählen Sie Installationsmöglichkeiten.



4. Wählen Sie den Installationsort. Es wird empfohlen, die Software auf einem anderen Laufwerk als dem Systemlaufwerk zu installieren.

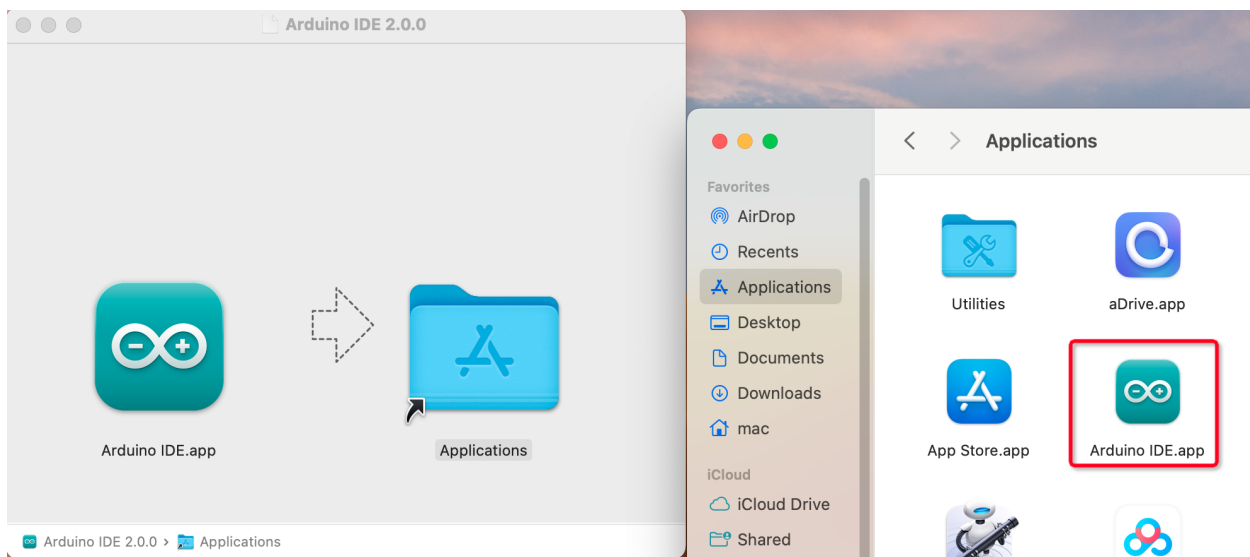


5. Dann Fertigstellen.



macOS

Doppelklicken Sie auf die heruntergeladene Datei `arduino_ide_XXXX.dmg` und folgen Sie den Anweisungen, um die **Arduino IDE.app** in den **Applications**-Ordner zu kopieren. Sie werden sehen, dass die Arduino IDE nach einigen Sekunden erfolgreich installiert ist.

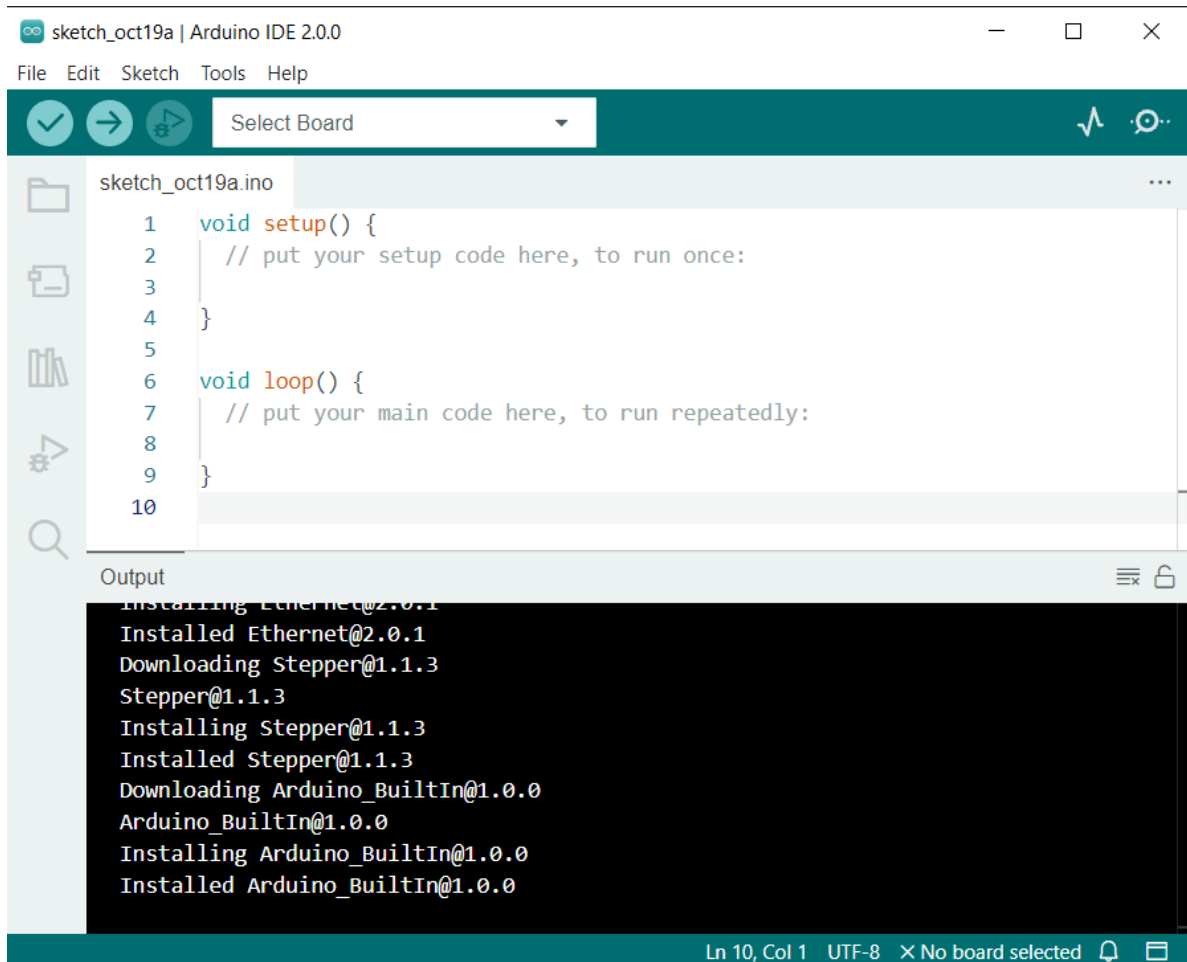


Linux

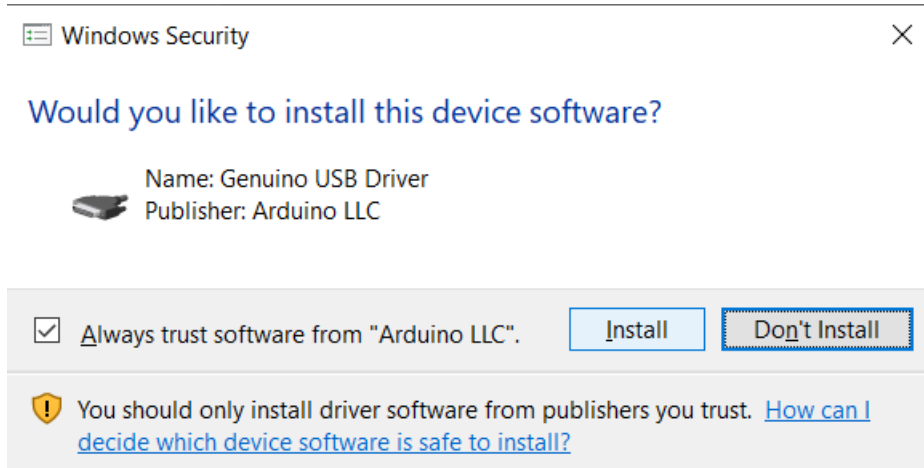
Für das Tutorial zur Installation der Arduino IDE 2.0 auf einem Linux-System, siehe bitte

Öffnen der IDE

1. Wenn Sie die Arduino IDE 2.0 zum ersten Mal öffnen, installiert sie automatisch die Arduino AVR Boards, eingebaute Bibliotheken und andere erforderliche Dateien.



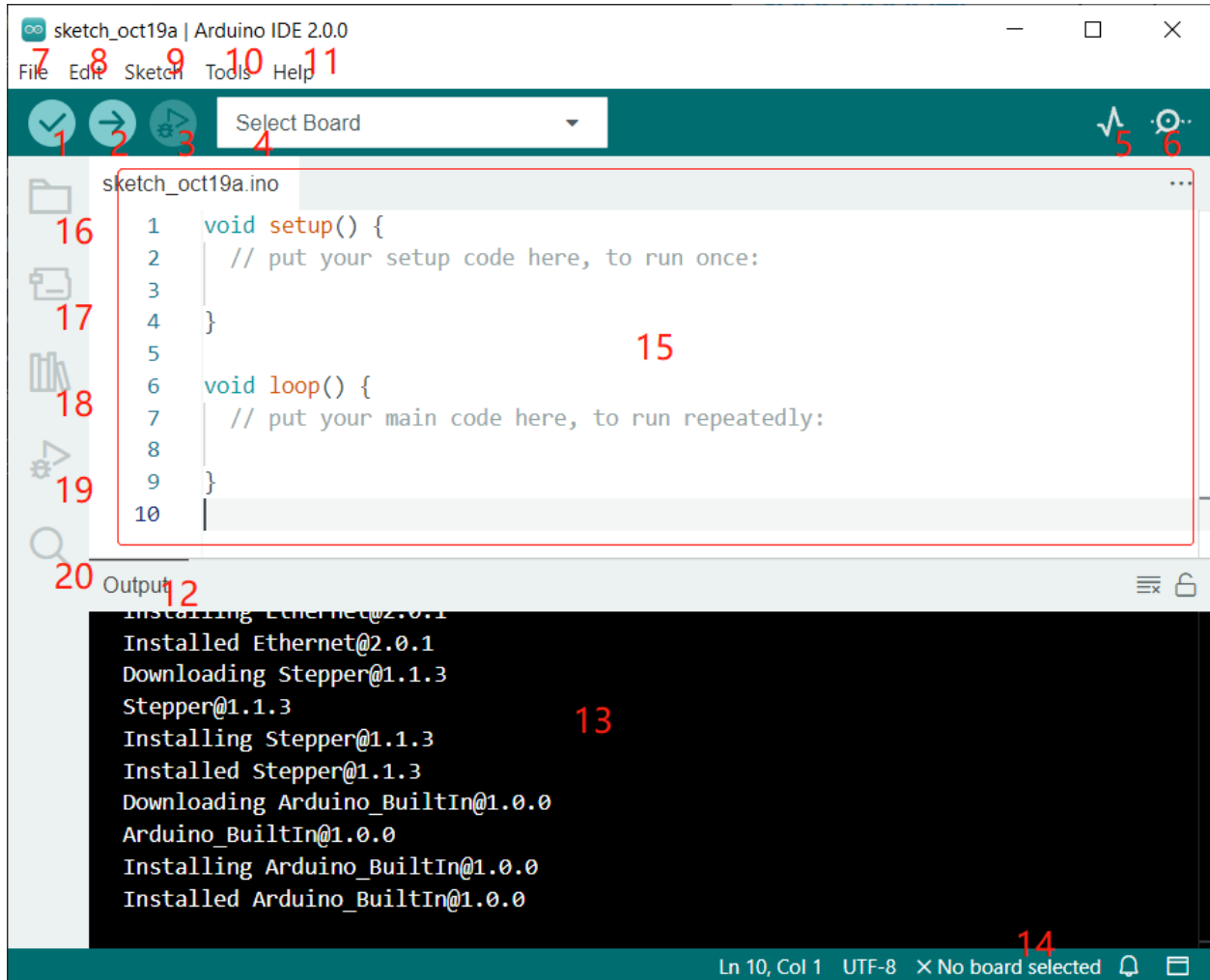
2. Zusätzlich kann Ihr Firewall- oder Sicherheitscenter einige Male aufpoppen und fragen, ob Sie einige Gerätetreiber installieren möchten. Bitte installieren Sie alle davon.



3. Jetzt ist Ihre Arduino IDE einsatzbereit!

Bemerkung: Falls einige Installationen aufgrund von Netzwerkproblemen oder anderen Gründen nicht funktioniert haben, können Sie die Arduino IDE erneut öffnen und sie wird den Rest der Installation abschließen. Das Ausgabefenster wird nach Abschluss aller Installationen nicht automatisch geöffnet, es sei denn, Sie klicken auf Überprüfen oder Hochladen.

2.3.2 Einführung in die Arduino IDE

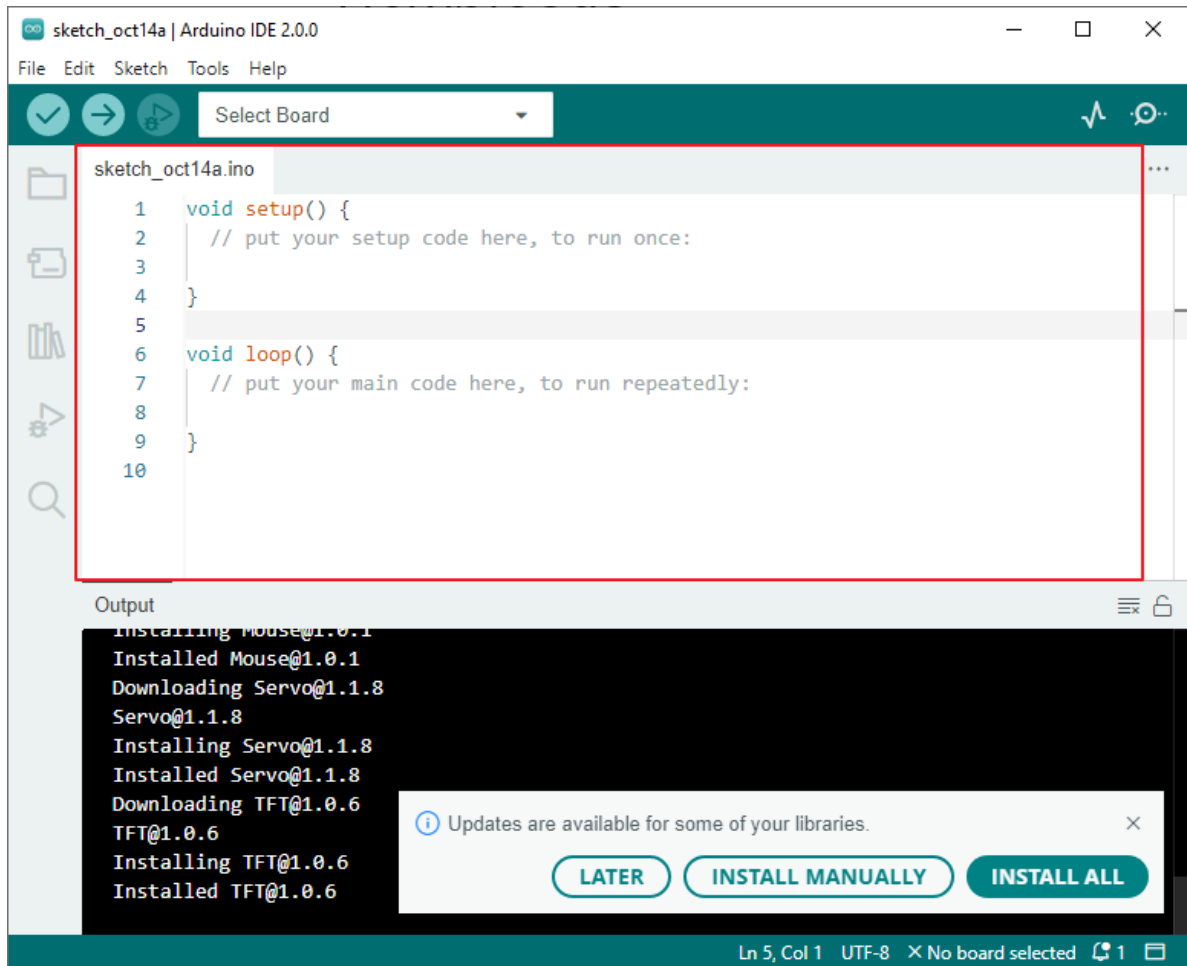


1. **Verify:** Kompilieren Sie Ihren Code. Etwaige Syntaxprobleme werden mit Fehlern angezeigt.
2. **Upload:** Laden Sie den Code auf Ihr Board. Wenn Sie den Knopf drücken, flackern die RX- und TX-LEDs auf dem Board schnell und hören nicht auf, bis das Hochladen abgeschlossen ist.
3. **Debug:** Für die zeilenweise Fehlerprüfung.
4. **Select Board:** Schnelleinrichtung von Board und Port.
5. **Serial Plotter:** Überprüfen Sie die Änderungen des Messwerts.
6. **Serial Monitor:** Klicken Sie auf den Knopf und ein Fenster erscheint. Es empfängt die Daten, die von Ihrem Steuerungsboard gesendet werden. Dies ist sehr nützlich für das Debugging.
7. **File:** Klicken Sie auf das Menü und eine Dropdown-Liste erscheint, einschließlich Dateierstellung, Öffnen, Speichern, Schließen, einige Parameterkonfigurationen usw.
8. **Edit:** Klicken Sie auf das Menü. In der Dropdown-Liste finden Sie Bearbeitungsvorgänge wie **Cut**, **Copy**, **Paste**, **Find** usw. mit ihren entsprechenden Tastenkürzeln.
9. **Sketch:** Beinhaltet Vorgänge wie **Verify**, **Upload**, **Add** usw. Eine wichtige Funktion ist **Include Library** – hier können Sie Bibliotheken hinzufügen.

10. **Werkzeug:** Beinhaltet einige Werkzeuge – am häufigsten verwendet werden Board (das Board, das Sie verwenden) und Port (der Port, an dem Ihr Board angeschlossen ist). Jedes Mal, wenn Sie den Code hochladen möchten, müssen Sie diese auswählen oder überprüfen.
11. **Tool:** Wenn Sie Anfänger sind, können Sie die Optionen im Menü überprüfen und die benötigte Hilfe erhalten, einschließlich Operationen in der IDE, Einführungsinformationen, Fehlerbehebung, Codeerklärung usw.
12. **Output Bar:** Wechseln Sie hier zur Ausgabekarte.
13. **Output Window:** Druckt Informationen.
14. **Board and Port:** Hier können Sie das für das Hochladen des Codes ausgewählte Board und den Port anzeigen. Sie können sie erneut über **Tools** -> **Board / Port** auswählen, falls etwas nicht stimmt.
15. Der Bearbeitungsbereich der IDE. Hier können Sie Code schreiben.
16. **Sketchbook:** Zum Verwalten von Sketch-Dateien.
17. **Board Manager:** Zum Verwalten von Board-Treibern.
18. **Library Manager:** Zum Verwalten Ihrer Bibliotheksdateien.
19. **Debug:** Hilft beim Debuggen von Code.
20. **Search:** Suchen Sie Codes aus Ihren Skizzen.

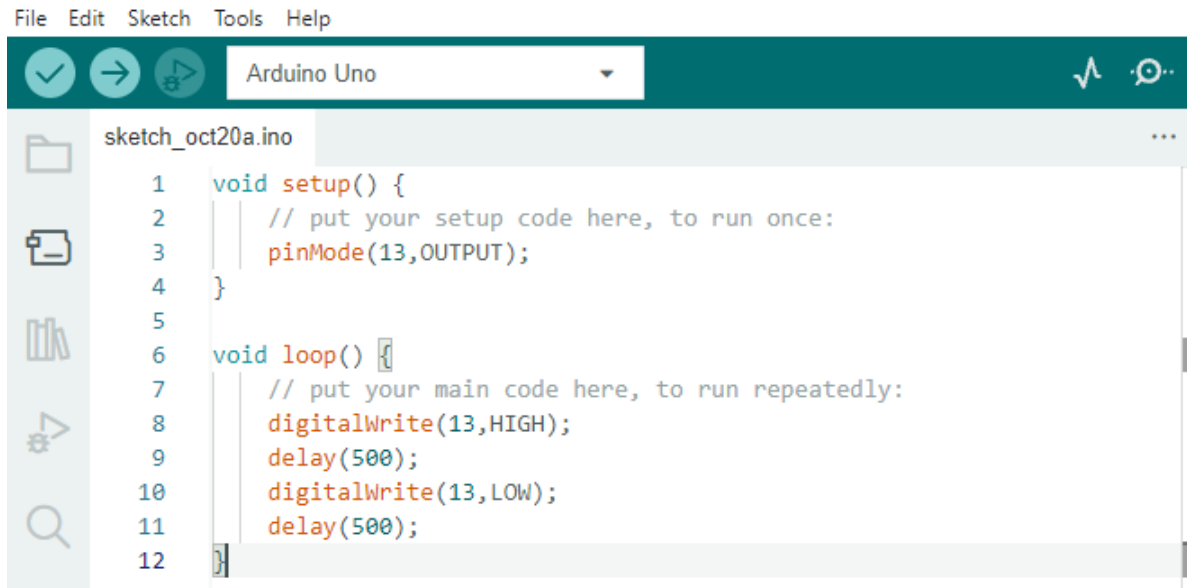
2.3.3 Wie man einen Sketch erstellt, öffnet oder speichert?

1. Wenn Sie die Arduino IDE zum ersten Mal öffnen oder einen neuen Sketch erstellen, sehen Sie eine Seite wie diese, auf der die Arduino IDE eine neue Datei für Sie erstellt, die als „Sketch“ bezeichnet wird.



Diese Sketch-Dateien haben einen regulären temporären Namen, an dem Sie das Erstellungsdatum der Datei erkennen können. `sketch_oct14a.ino` bedeutet erster Sketch am 14. Oktober, `.ino` ist das Dateiformat dieses Sketches.

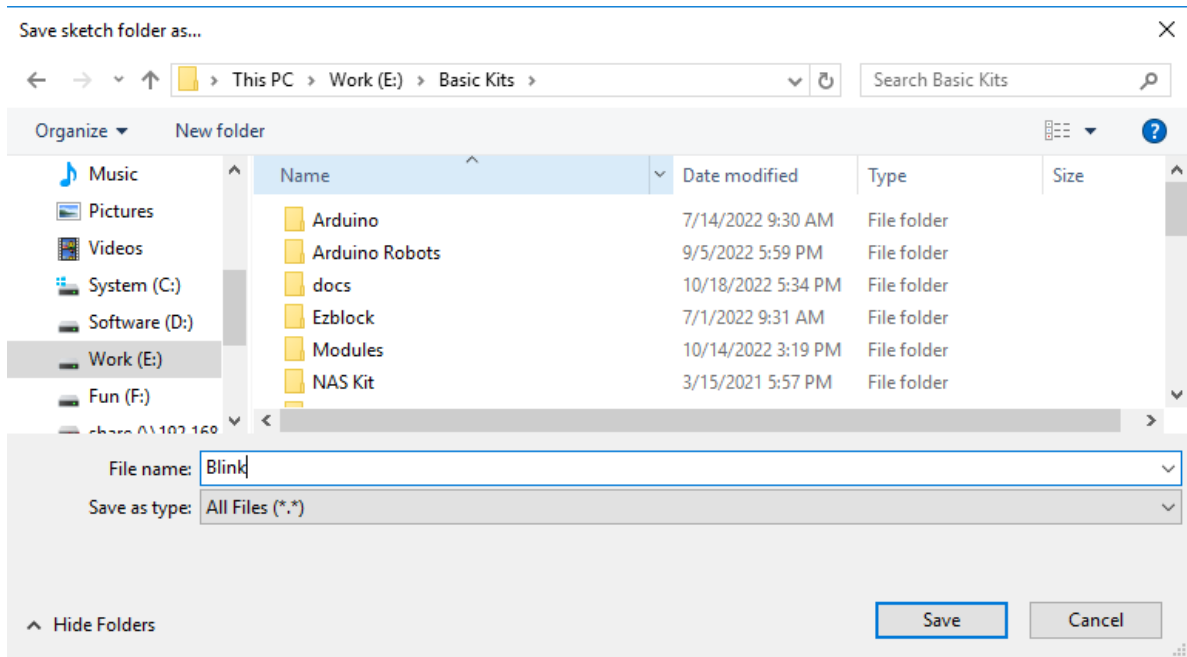
2. Jetzt versuchen wir, einen neuen Sketch zu erstellen. Kopieren Sie den folgenden Code in die Arduino IDE, um den ursprünglichen Code zu ersetzen.



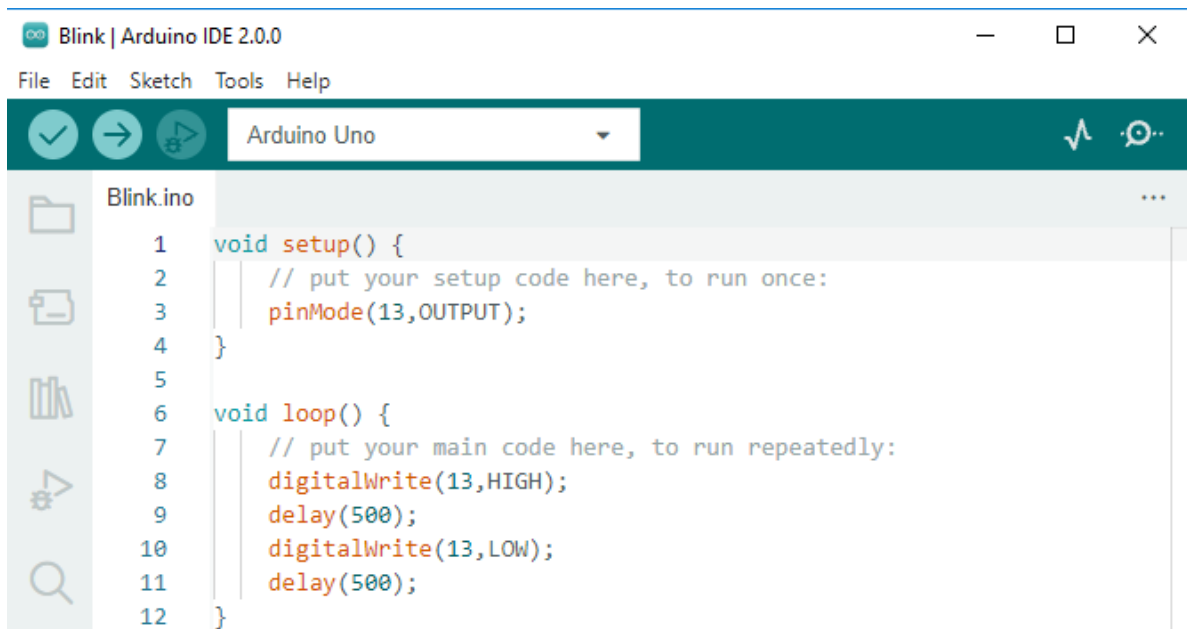
```
void setup() {
    // put your setup code here, to run once:
    pinMode(13,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(13,HIGH);
    delay(500);
    digitalWrite(13,LOW);
    delay(500);
}
```

3. Drücken Sie Ctrl+S oder klicken Sie auf **File** -> **Save**. Der Sketch wird standardmäßig unter C:\Users\{your_user}\Documents\Arduino gespeichert. Sie können ihn umbenennen oder einen neuen Pfad zum Speichern auswählen.



4. Nach erfolgreichem Speichern sehen Sie, dass der Name in der Arduino IDE aktualisiert wurde.



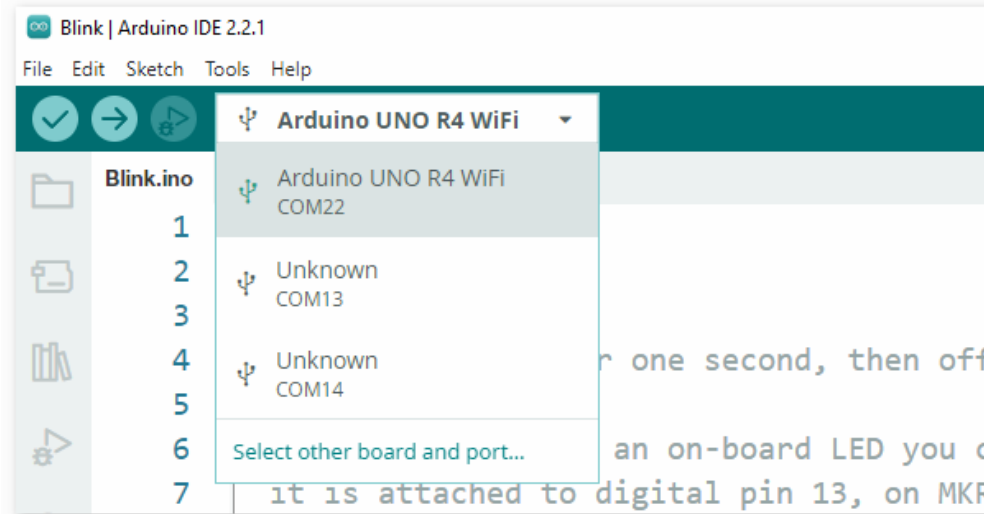
2.3.4 Wie man einen Sketch auf das Board hochlädt?

In diesem Abschnitt lernen Sie, wie Sie den zuvor erstellten Sketch auf das Arduino-Board hochladen, sowie einige wichtige Überlegungen.

1. Wählen Sie Board und Port aus

Arduino-Entwicklungsboards werden normalerweise mit einem USB-Kabel geliefert. Sie können es verwenden, um das Board mit Ihrem Computer zu verbinden.

Wählen Sie das richtige **Board** und den richtigen **Port** in der Arduino IDE aus. Normalerweise werden Arduino-Boards automatisch vom Computer erkannt und einem Port zugewiesen, sodass Sie ihn hier auswählen können.



Wenn Ihr Board bereits angeschlossen ist, aber nicht erkannt wird, überprüfen Sie, ob das **INSTALLED**-Logo im Abschnitt **Arduino UNO R4 Boards** des **Boards Manager** erscheint. Wenn nicht, scrollen Sie bitte etwas nach unten und klicken Sie auf **INSTALL**.

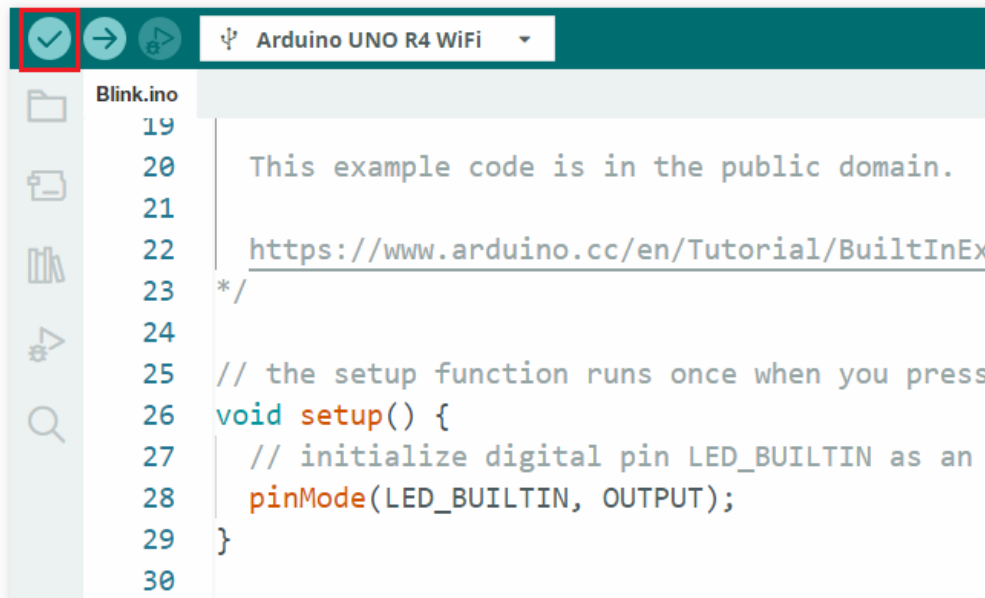
Suchen Sie „UNO R4“ im **Boards Manager** und überprüfen Sie, ob die entsprechende Bibliothek installiert ist.



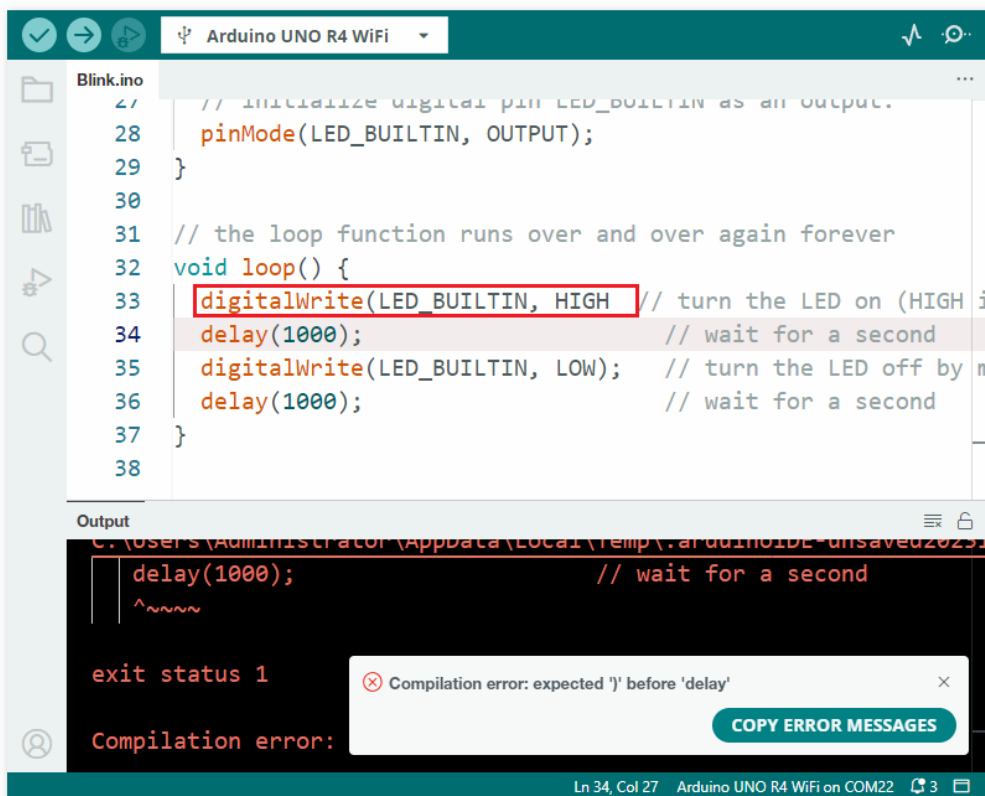
Das Neueröffnen der Arduino IDE und das erneute Anschließen des Arduino-Boards beheben die meisten Probleme. Sie können auch **Tools -> Board** oder **Port** anklicken, um sie auszuwählen.

2. Überprüfen Sie den Sketch

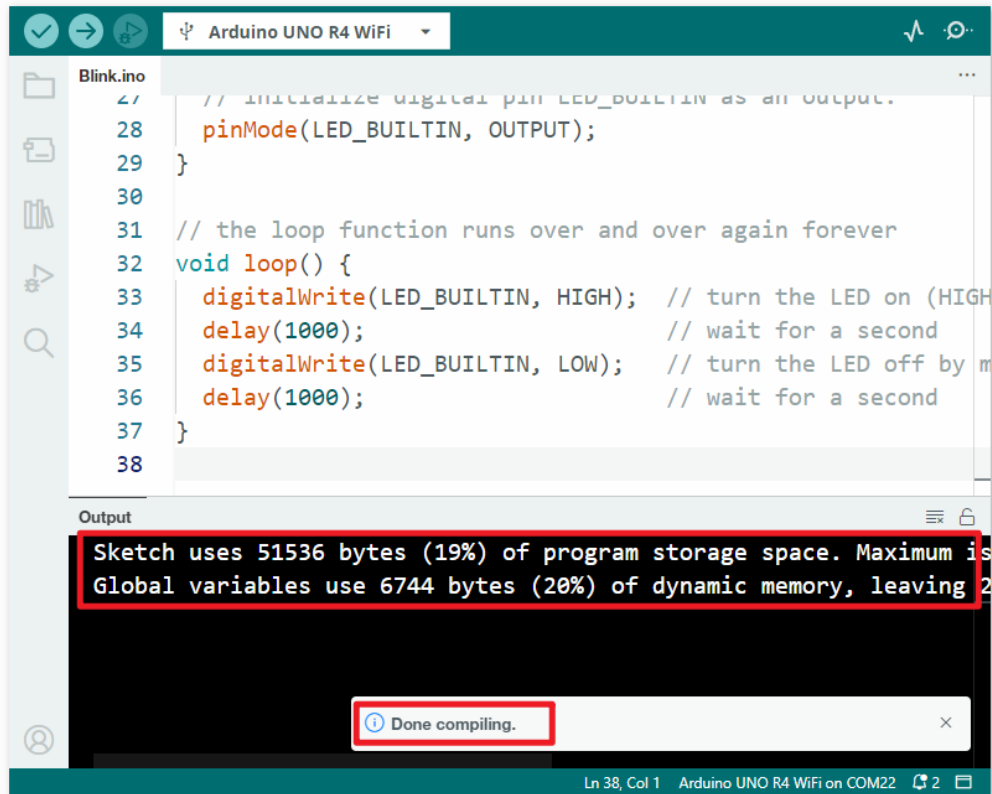
Nachdem Sie auf die Schaltfläche Überprüfen geklickt haben, wird der Sketch kompiliert, um zu sehen, ob Fehler vorliegen.



Sie können es verwenden, um Fehler zu finden, wenn Sie einige Zeichen löschen oder versehentlich einige Buchstaben eingeben. In der Nachrichtenleiste können Sie sehen, wo und welche Art von Fehlern aufgetreten sind.

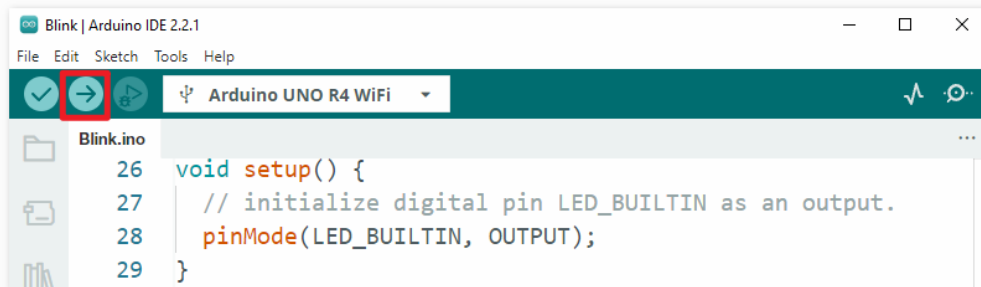


Wenn keine Fehler vorhanden sind, sehen Sie eine Nachricht wie die folgende.

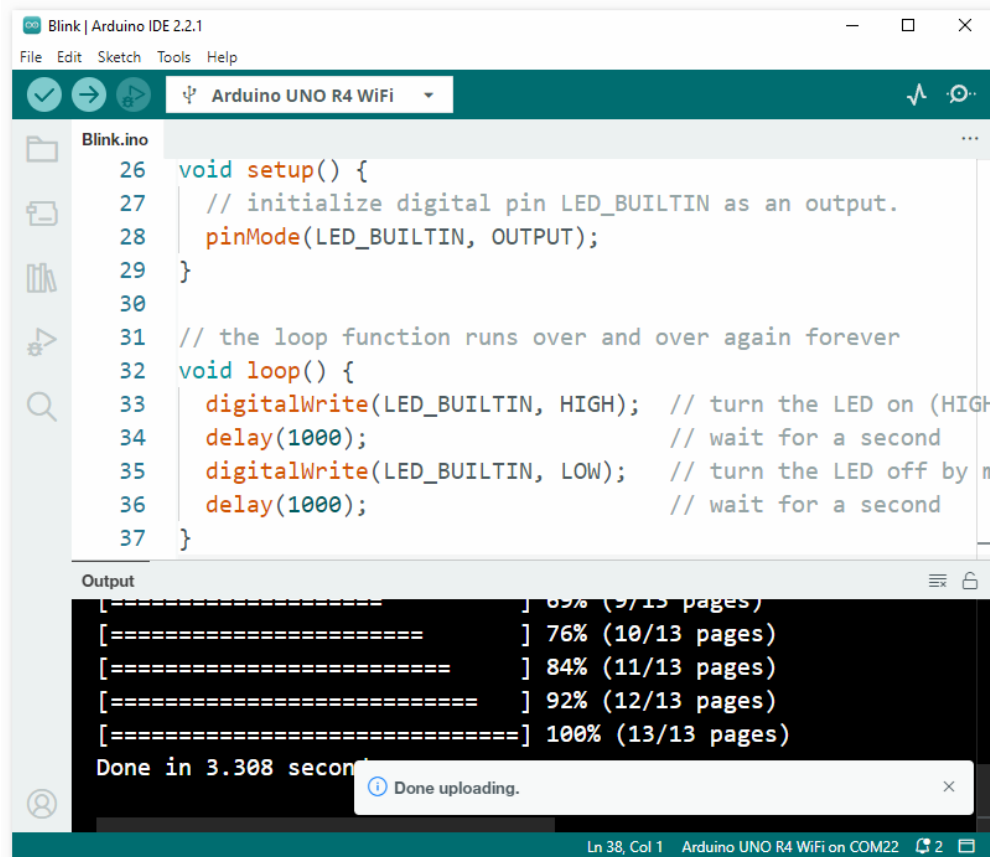


3. Sketch hochladen

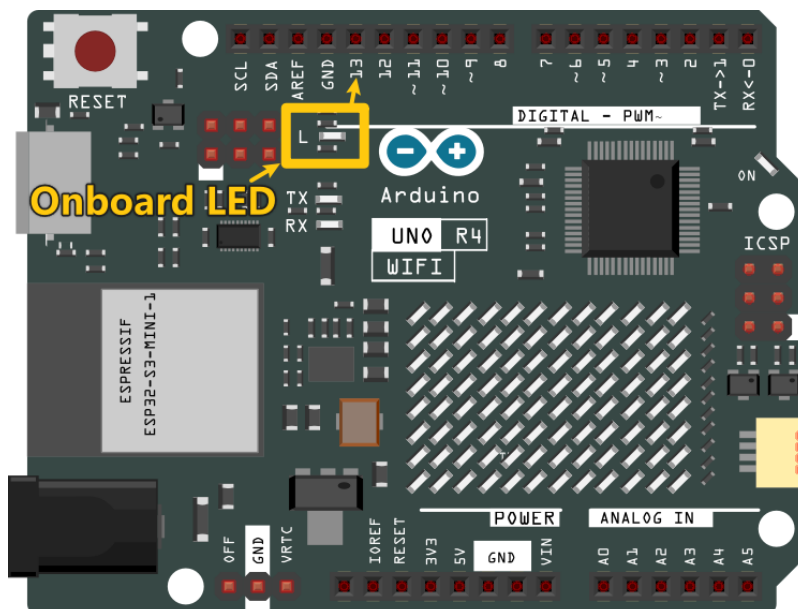
Nachdem Sie die obigen Schritte abgeschlossen haben, klicken Sie auf die Schaltfläche **Upload**, um diesen Sketch auf das Board zu laden.



Wenn es erfolgreich ist, können Sie die folgende Aufforderung sehen.



Gleichzeitig blinkt die LED auf dem Board.



Das Arduino-Board führt den Sketch automatisch aus, nachdem er hochgeladen wurde und das Board mit Strom versorgt wird. Das laufende Programm kann durch Hochladen eines neuen Sketches überschrieben werden.

2.3.5 Arduino Programmstruktur

Schauen wir uns die neue Sketch-Datei an. Obwohl sie selbst einige Zeilen Code enthält, ist es tatsächlich ein „leerer“ Sketch. Das Hochladen dieses Sketches auf das Entwicklungsboard bewirkt nichts.

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Wenn wir `setup()` und `loop()` entfernen und den Sketch zu einer wirklich leeren Datei machen, werden Sie feststellen, dass er die Überprüfung nicht besteht. Sie sind gleichbedeutend mit dem menschlichen Skelett und unverzichtbar.

Während des Skizzierens wird `setup()` zuerst ausgeführt, und der Code darin (innerhalb von `{}`) wird ausgeführt, nachdem das Board mit Strom versorgt oder zurückgesetzt wurde, und zwar nur einmal. `loop()` wird verwendet, um die Hauptfunktion zu schreiben, und der Code darin wird in einer Schleife ausgeführt, nachdem `setup()` ausgeführt wurde.

Um `setup()` und `loop()` besser zu verstehen, verwenden wir vier Skizzen. Ihr Zweck ist es, die LED auf dem Arduino-Board blinken zu lassen. Bitte führen Sie jedes Experiment nacheinander durch und zeichnen Sie ihre spezifischen Effekte auf.

- Sketch 1: Lassen Sie die LED auf dem Board kontinuierlich blinken.

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}
```

- Sketch 2: Lassen Sie die LED auf dem Board nur einmal blinken.

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

- Sketch 3: Lassen Sie die LED auf dem Board langsam einmal blinken und dann schnell blinken.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
    digitalWrite(13,HIGH);  
    delay(1000);  
    digitalWrite(13,LOW);  
    delay(1000);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,HIGH);  
    delay(200);  
    digitalWrite(13,LOW);  
    delay(200);  
}
```

- Sketch 4: Meldet einen Fehler.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
digitalWrite(13,HIGH);  
delay(1000);  
digitalWrite(13,LOW);  
delay(1000);  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

Mit Hilfe dieser Skizzen können wir mehrere Merkmale von `setup-loop` zusammenfassen.

- `loop()` wird wiederholt ausgeführt, nachdem das Board mit Strom versorgt wurde.
- `setup()` wird nur einmal ausgeführt, nachdem das Board mit Strom versorgt wurde.
- Nachdem das Board mit Strom versorgt wurde, wird zuerst `setup()` ausgeführt, gefolgt von `loop()`.
- Der Code muss innerhalb des `{}`-Bereichs von `setup()` oder `loop()` geschrieben werden, außerhalb des Rahmens wird ein Fehler auftreten.

Bemerkung: Anweisungen wie `digitalWrite(13,HIGH)` werden verwendet, um die On-Board-LED anzusteuern, und wir werden in späteren Kapiteln detailliert auf ihre Verwendung eingehen.

2.3.6 Regeln für das Schreiben eines Sketches

Wenn Sie einen Freund bitten, das Licht für Sie einzuschalten, können Sie sagen „Schalte das Licht ein.“ oder „Licht an, Kumpel.“, Sie können jeden Tonfall verwenden, den Sie möchten.

Wenn Sie jedoch möchten, dass das Arduino-Board etwas für Sie tut, müssen Sie die Programmiersprache von Arduino befolgen, um die Befehle einzugeben.

Dieses Kapitel enthält die grundlegenden Regeln der Arduino-Sprache und wird Ihnen helfen zu verstehen, wie man natürliche Sprache in Code übersetzt.

Natürlich ist dies ein Prozess, der Zeit braucht, um sich daran zu gewöhnen, und es ist auch der fehleranfälligste Teil des Prozesses für Anfänger. Wenn Sie also oft Fehler machen, ist das in Ordnung, versuchen Sie es einfach noch ein paar Mal.

Semikolon ;

Genau wie beim Schreiben eines Briefes, wo Sie am Ende jedes Satzes einen Punkt als Ende schreiben, erfordert die Arduino-Sprache, dass Sie ; verwenden, um dem Board das Ende des Befehls mitzuteilen.

Nehmen wir das bekannte Beispiel „Blinkende LED an Bord“. Ein korrekter Sketch sollte so aussehen.

Beispiel:

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}
```

Als Nächstes werfen wir einen Blick auf die folgenden zwei Skizzen und raten, ob sie vor dem Ausführen von Arduino korrekt erkannt werden können.

Sketch A:

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH)
  delay(500)
  digitalWrite(13,LOW)
  delay(500)
}
```

Sketch B:

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,  
HIGH); delay  
    (500  
    );  
    digitalWrite(13,  
  
    LOW);  
        delay(500)  
    ;  
}
```

Das Ergebnis ist, dass **Sketch A** einen Fehler meldet und **Sketch B** funktioniert.

- Die Fehler in **Sketch A** sind fehlende ; und obwohl es normal aussieht, kann Arduino es nicht lesen.
- **Sketch B**, sieht ungewöhnlich aus, aber tatsächlich existieren Einrückungen, Zeilenumbrüche und Leerzeichen in Arduino-Programmen nicht, also sieht es für den Arduino-Compiler genauso aus wie im Beispiel.

Schreiben Sie Ihren Code jedoch bitte nicht wie **Sketch B**, denn normalerweise sind es Menschen, die den Code schreiben und betrachten, also bringen Sie sich nicht in Schwierigkeiten.

Geschweifte Klammern {}

{ } ist ein Hauptbestandteil der Arduino-Programmiersprache, und sie müssen paarweise auftreten. Eine bessere Programmierkonvention ist es, eine Struktur, die geschweifte Klammern erfordert, einzufügen, indem man die rechte geschweifte Klammer direkt nach dem Tippen der linken geschweiften Klammer eintippt und dann den Cursor zwischen die Klammern bewegt, um die Anweisung einzufügen.

Kommentar //

Kommentare sind der Teil des Sketches, den der Compiler ignoriert. Sie werden normalerweise verwendet, um anderen zu erklären, wie das Programm funktioniert.

Wenn wir zwei benachbarte Schrägstriche in einer Codezeile schreiben, ignoriert der Compiler alles bis zum Ende der Zeile.

Wenn wir einen neuen Sketch erstellen, kommt er mit zwei Kommentaren, und wenn wir diese zwei Kommentare entfernen, wird der Sketch in keiner Weise beeinflusst.

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

}

Kommentare sind sehr nützlich in der Programmierung, und einige häufige Verwendungen sind unten aufgeführt.

- Verwendung A: Sich selbst oder anderen erklären, was dieser Abschnitt des Codes macht.

```
void setup() {
  pinMode(13,OUTPUT); //Set pin 13 to output mode, it controls the onboard LED
}

void loop() {
  digitalWrite(13,HIGH); // Activate the onboard LED by setting pin 13 high
  delay(500); // Status quo for 500 ms
  digitalWrite(13,LOW); // Turn off the onboard LED
  delay(500); // Status quo for 500 ms
}
```

- Verwendung B: Einige Aussagen vorübergehend invalidieren (ohne sie zu löschen) und sie entsperren, wenn Sie sie verwenden müssen, sodass Sie sie nicht neu schreiben müssen. Dies ist sehr nützlich beim Debuggen von Code und beim Versuch, Programmfehler zu lokalisieren.

```
void setup() {
  pinMode(13,OUTPUT);
  // digitalWrite(13,HIGH);
  // delay(1000);
  // digitalWrite(13,LOW);
  // delay(1000);
}

void loop() {
  digitalWrite(13,HIGH);
  delay(200);
  digitalWrite(13,LOW);
  delay(200);
}
```

Bemerkung: Verwenden Sie das Tastenkürzel Ctrl+/, um Ihnen beim schnellen Kommentieren oder Entkommentieren Ihres Codes zu helfen.

Kommentar /**/

Wie // für Kommentare. Diese Art von Kommentar kann mehr als eine Zeile lang sein, und sobald der Compiler /* liest, ignoriert er alles, was folgt, bis er auf */ trifft.

Beispiel 1:

```
/* Blink */

void setup() {
  pinMode(13,OUTPUT);
```

(Fortsetzung auf der nächsten Seite)

```
}  
  
void loop() {  
    /*  
    The following code will blink the onboard LED  
    You can modify the number in delay() to change the blinking frequency  
    */  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}
```

#define

Dies ist ein nützliches Werkzeug in C++.

```
#define identifier token-string
```

Der Compiler ersetzt automatisch `identifier` durch `token-string`, wenn er es liest, was normalerweise für Konstantendefinitionen verwendet wird.

Als Beispiel hier ein Sketch, der `define` verwendet, was die Lesbarkeit des Codes verbessert.

```
#define ONBOARD_LED 13  
#define DELAY_TIME 500  
  
void setup() {  
    pinMode(ONBOARD_LED,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(ONBOARD_LED,HIGH);  
    delay(DELAY_TIME);  
    digitalWrite(ONBOARD_LED,LOW);  
    delay(DELAY_TIME);  
}
```

Für den Compiler sieht es tatsächlich so aus.

```
void setup() {  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}
```

Wir können sehen, dass der `identifier` ersetzt wird und im Programm nicht existiert. Daher gibt es einige Vorsichtsmaßnahmen bei der Verwendung.

1. Ein token-string kann nur manuell geändert werden und kann nicht durch Rechenoperationen im Programm in andere Werte umgewandelt werden.
2. Vermeiden Sie die Verwendung von Symbolen wie ;. Zum Beispiel.

```
#define ONBOARD_LED 13;

void setup() {
    pinMode(ONBOARD_LED,OUTPUT);
}

void loop() {
    digitalWrite(ONBOARD_LED,HIGH);
}
```

Der Compiler wird es als Folgendes erkennen, was als Fehler gemeldet wird.

```
void setup() {
    pinMode(13;,OUTPUT);
}

void loop() {
    digitalWrite(13;,HIGH);
}
```

Bemerkung: Eine Namenskonvention für #define ist, den identifier zu groß zu schreiben, um Verwechslungen mit Variablen zu vermeiden.

2.3.7 Variable

Die Variable ist eines der mächtigsten und kritischsten Werkzeuge in einem Programm. Sie hilft uns, Daten in unseren Programmen zu speichern und abzurufen.

Das folgende Sketch-File verwendet Variablen. Es speichert die Pin-Nummern der LED an Bord in der Variablen ledPin und eine Zahl „500“ in der Variablen delayTime.

```
int ledPin = 13;
int delayTime = 500;

void setup() {
    pinMode(ledPin,OUTPUT);
}

void loop() {
    digitalWrite(ledPin,HIGH);
    delay(delayTime);
    digitalWrite(ledPin,LOW);
    delay(delayTime);
}
```

Warte, ist das eine Duplikation dessen, was #define macht? Die Antwort ist NEIN.

- Die Rolle von `#define` ist es, einfach und direkt Text zu ersetzen, es wird vom Compiler nicht als Teil des Programms betrachtet.
- Eine Variable hingegen existiert innerhalb des Programms und wird verwendet, um Werte zu speichern und abzurufen. Eine Variable kann auch ihren Wert innerhalb des Programms ändern, was ein `define` nicht kann.

Das untenstehende Sketch-File fügt sich selbst zur Variablen hinzu und wird dazu führen, dass die LED an Bord nach jedem Blinken länger blinkt.

```
int ledPin = 13;
int delayTime = 500;

void setup() {
  pinMode(ledPin,OUTPUT);
}

void loop() {
  digitalWrite(ledPin,HIGH);
  delay(delayTime);
  digitalWrite(ledPin,LOW);
  delay(delayTime);
  delayTime = delayTime+200; //Each execution increments the value by 200
}
```

Eine Variable deklarieren

Eine Variable zu deklarieren bedeutet, eine Variable zu erstellen.

Um eine Variable zu deklarieren, benötigen Sie zwei Dinge: den Datentyp und den Variablennamen. Der Datentyp muss durch ein Leerzeichen von der Variable getrennt sein, und die Variablendeklaration muss mit einem `;` beendet werden.

Verwenden wir diese Variable als Beispiel.

```
int delayTime;
```

Datentyp

Hier ist `int` ein Datentyp, der als Integer-Typ bezeichnet wird und verwendet werden kann, um ganze Zahlen von -32768 bis 32766 zu speichern. Es kann jedoch keine Dezimalzahlen speichern.

Variablen können andere Arten von Daten als ganze Zahlen speichern. Die Arduino-Sprache (die, wie wir uns erinnern, C++ ist) hat eingebaute Unterstützung für einige davon (nur die am häufigsten verwendeten und nützlichsten sind hier aufgeführt):

- `float`: Speichert eine Dezimalzahl, zum Beispiel 3,1415926.
- `byte`: Kann Zahlen von 0 bis 255 speichern.
- `boolean`: Hält nur zwei mögliche Werte, `True` oder `False`, obwohl es ein Byte im Speicher belegt.
- `char`: Hält eine Zahl von -127 bis 127. Da es als `char` gekennzeichnet ist, versucht der Compiler, es einer Zeichen aus dem zuzuordnen.
- `string`: Kann eine Zeichenkette speichern, z.B. Halloween.

Variablenname

Sie können der Variablen jeden Namen geben, den Sie möchten, wie `i`, `apple`, `Bruce`, `R2D2`, `Sectumsempra`, aber es gibt einige grundlegende Regeln zu befolgen.

1. Beschreiben Sie, wofür sie verwendet wird. Hier habe ich die Variable `delayTime` genannt, damit Sie leicht verstehen können, was sie macht. Es funktioniert auch, wenn ich die Variable `barryAllen` nenne, aber es verwirrt die Person, die den Code betrachtet.
2. Verwenden Sie eine reguläre Nomenklatur. Sie können CamelCase verwenden, wie ich es getan habe, mit dem anfänglichen T in `delayTime`, damit leicht zu erkennen ist, dass die Variable aus zwei Wörtern besteht. Sie können auch UnderScoreCase verwenden, um die Variable als `delay_time` zu schreiben. Es hat keinen Einfluss auf den Programmablauf, aber es würde dem Programmierer helfen, den Code zu lesen, wenn Sie die von Ihnen bevorzugte Nomenklatur verwenden.
3. Verwenden Sie keine Schlüsselwörter. Ähnlich wie bei der Eingabe von „int“ färbt die Arduino-IDE es, um Sie daran zu erinnern, dass es ein Wort mit einer besonderen Bedeutung ist und nicht als Variablenname verwendet werden kann. Ändern Sie den Namen der Variablen, wenn sie gefärbt ist.
4. Spezielle Symbole sind nicht erlaubt. Zum Beispiel Raum, #, \$, /, +, % usw. Die Kombination aus englischen Buchstaben (Groß-/Kleinschreibung beachten), Unterstrichen und Zahlen (aber Zahlen können nicht als erstes Zeichen eines Variablennamens verwendet werden) ist reichhaltig genug.

Einer Variablen einen Wert zuweisen

Sobald wir die Variable deklariert haben, ist es an der Zeit, die Daten zu speichern. Wir verwenden den Zuweisungsoperator (d.h. =), um den Wert in die Variable zu setzen.

Wir können Werte der Variablen zuweisen, sobald wir sie deklarieren.

```
int delayTime = 500;
```

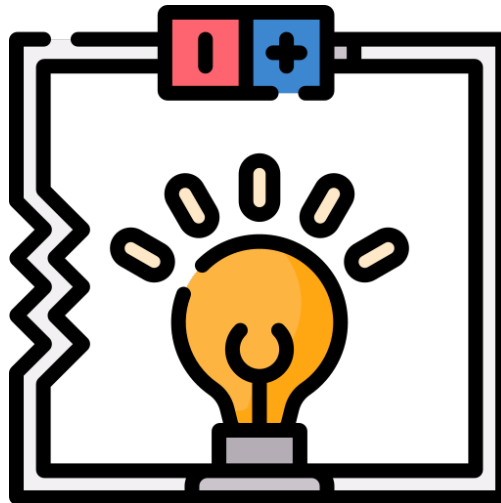
Es ist auch möglich, ihr zu einem späteren Zeitpunkt einen neuen Wert zuzuweisen.

```
int delayTime; // no value
delayTime = 500; // value is 500
delayTime = delayTime + 200; // value is 700
```

2.3.8 Wie man den Schaltkreis aufbaut

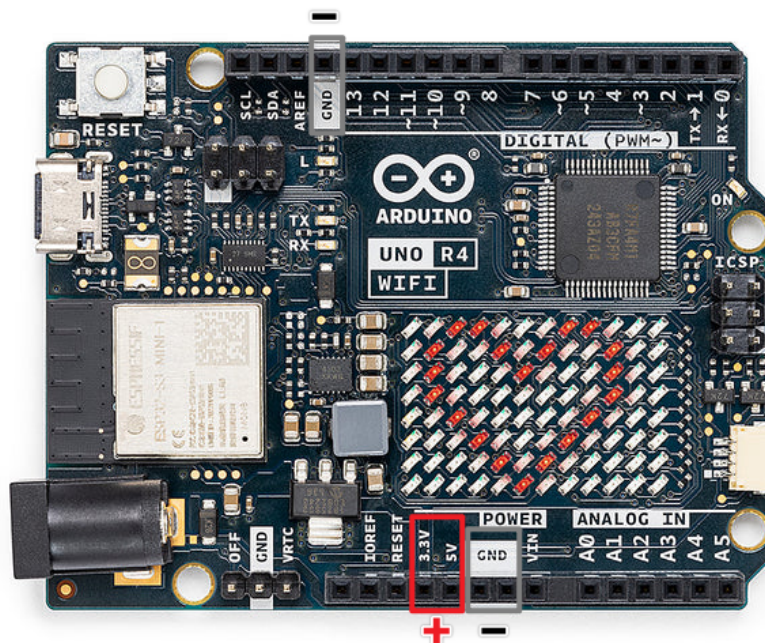
Viele Dinge, die Sie täglich verwenden, werden mit Strom betrieben, wie die Lichter in Ihrem Haus und der Computer, auf dem Sie gerade lesen.

Um Elektrizität zu nutzen, müssen Sie einen elektrischen Schaltkreis aufbauen. Ein Schaltkreis ist grundsätzlich ein Weg, durch den Elektrizität fließt, oder ein elektronischer Schaltkreis, und besteht aus elektrischen Geräten und Komponenten (Geräten), die auf bestimmte Weise verbunden sind, wie Widerstände, Kondensatoren, Stromversorgungen und Schalter.



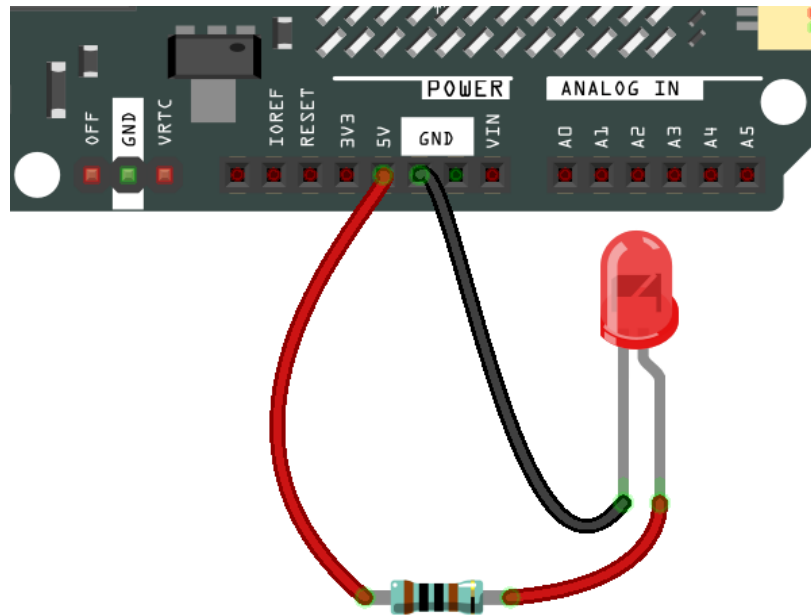
Ein Schaltkreis ist ein geschlossener Weg, in dem Elektronen sich bewegen, um einen elektrischen Strom zu erzeugen. Damit Strom fließen kann, muss es einen leitenden Pfad zwischen dem positiven Anschluss der Stromversorgung und dem negativen Anschluss geben, was als geschlossener Schaltkreis bezeichnet wird (wenn er unterbrochen ist, spricht man von einem offenen Schaltkreis).

Das Arduino-Board hat einige Stromausgangspins (positiv) und einige Erdungspins (negativ). Sie können diese Pins als positive und negative Seiten der Stromversorgung nutzen, indem Sie die Stromquelle in das Board einstecken.



Mit Elektrizität können Sie Werke mit Licht, Klang und Bewegung erschaffen. Sie können eine LED zum Leuchten bringen, indem Sie den langen Pin an den positiven Anschluss und den kurzen Pin an den negativen Anschluss anschließen. Allerdings kann dies direkt gemacht schnell nicht nur die LED, sondern auch die Pins Ihres UNO R4-Boards beschädigen. Um dies zu vermeiden, ist es unerlässlich, einen 1k-Widerstand in den Schaltkreis einzufügen, um sowohl die LED als auch die Pins des UNO R4 zu schützen.

Der von ihnen gebildete Schaltkreis ist unten dargestellt.



Vielleicht haben Sie jetzt Fragen: Wie baue ich diesen Schaltkreis auf? Halte ich die Drähte mit der Hand fest oder klebe die Pins und Drähte?

In dieser Situation werden lötfreie Steckbretter Ihre stärksten Verbündeten sein.

Hallo, Steckbrett!

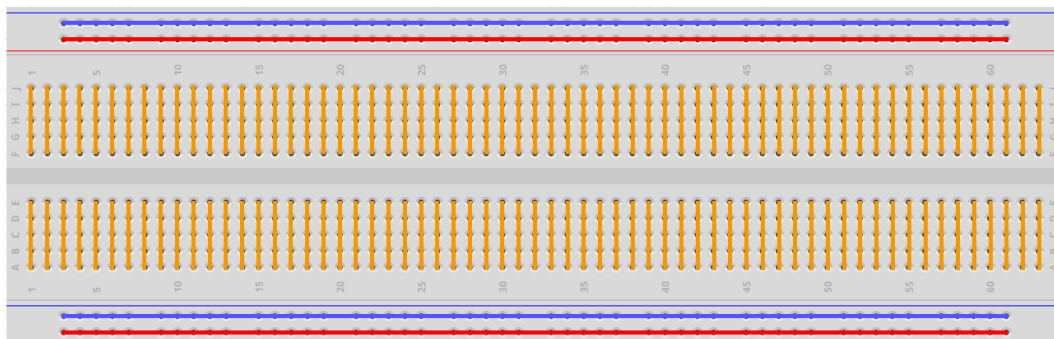
Ein Steckbrett ist eine rechteckige Kunststoffplatte mit einer Menge kleiner Löcher. Diese Löcher ermöglichen es uns, elektronische Komponenten leicht einzusetzen und elektronische Schaltkreise aufzubauen. Steckbretter fixieren elektronische Komponenten nicht dauerhaft, sodass wir einen Schaltkreis leicht reparieren und neu starten können, wenn etwas schiefgeht.

Bemerkung: Für die Verwendung von Steckbrettern sind keine speziellen Werkzeuge erforderlich. Viele elektronische Komponenten sind jedoch sehr klein, und eine Pinzette kann uns helfen, kleine Teile besser aufzuheben.

Im Internet finden wir viele Informationen über Steckbretter.

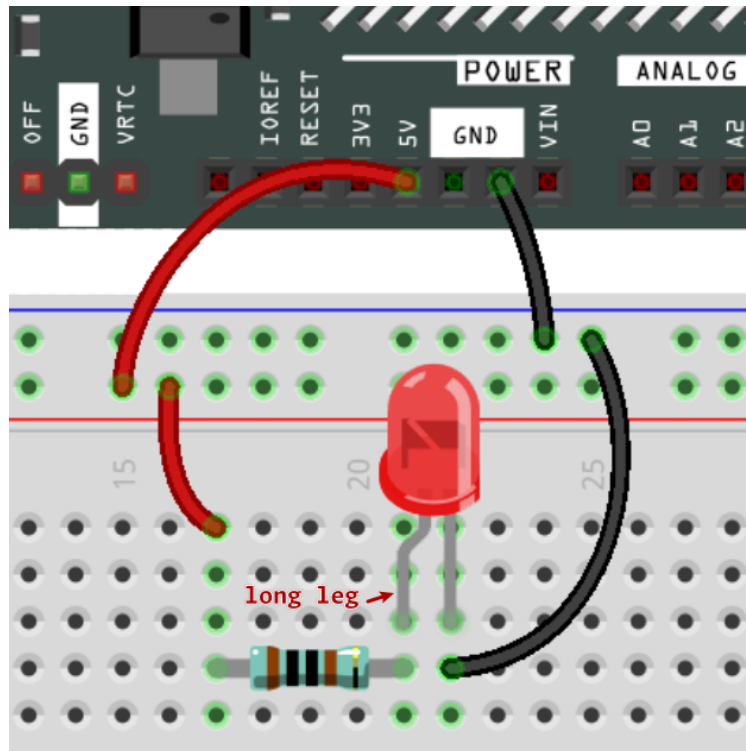
- [Wie man ein Steckbrett benutzt - Science Buddies](#)
- [Was ist ein STECKBRETT? - Makezine](#)

Hier sind einige Dinge, die Sie über Steckbretter wissen sollten.



1. Jede Halbzeilengruppe (wie Spalte A-E in Reihe 1 oder Spalte F-J in Reihe 3) ist verbunden. Wenn also ein elektrisches Signal bei A1 eingeht, kann es bei B1, C1, D1, E1 herausfließen, aber nicht bei F1 oder A2.
2. In den meisten Fällen werden beide Seiten des Steckbretts als Stromschienen verwendet, und die Löcher in jeder Spalte (etwa 50 Löcher) sind miteinander verbunden. Als allgemeine Regel gilt, dass positive Stromversorgungen an den Löchern in der Nähe des roten Drahts und negative Stromversorgungen an den Löchern in der Nähe des blauen Drahts angeschlossen werden.

Lassen Sie uns der Stromrichtung folgen, um den Schaltkreis aufzubauen!



1. In diesem Schaltkreis nutzen wir den 5V-Pin des Boards, um die LED mit Strom zu versorgen. Verwenden Sie ein männlich-zu-männliches (M2M) Jumperkabel, um ihn mit der roten Stromschiene zu verbinden.
2. Um die LED und die Pins des UNO R4 zu schützen, muss der Strom durch einen 1k-Ohm-Widerstand fließen. Verbinden Sie ein Ende (jedes Ende) des Widerstands mit der roten Stromschiene und das andere Ende mit einer freien Reihe des Steckbretts.

Bemerkung: Der Farbring des 1000-Ohm *Widerstand* ist rot, schwarz, schwarz, braun und braun.

3. Wenn Sie die LED aufnehmen, werden Sie sehen, dass einer ihrer Anschlüsse länger ist als der andere. Verbinden Sie den längeren Anschluss mit derselben Reihe wie der Widerstand und den kürzeren Anschluss mit einer anderen Reihe.

Bemerkung: Der längere Anschluss ist die Anode, die die positive Seite des Schaltkreises repräsentiert; der kürzere Anschluss ist die Kathode, die die negative Seite repräsentiert.

Die Anode muss über einen Widerstand mit dem GPIO-Pin verbunden sein; die Kathode muss mit dem GND-Pin verbunden sein.

4. Verwenden Sie ein männlich-zu-männliches (M2M) Jumperkabel, um den kurzen Pin der LED mit der negativen Stromschiene des Steckbretts zu verbinden.
5. Verbinden Sie den GND-Pin des Boards mit der negativen Stromschiene mit einem Jumper.

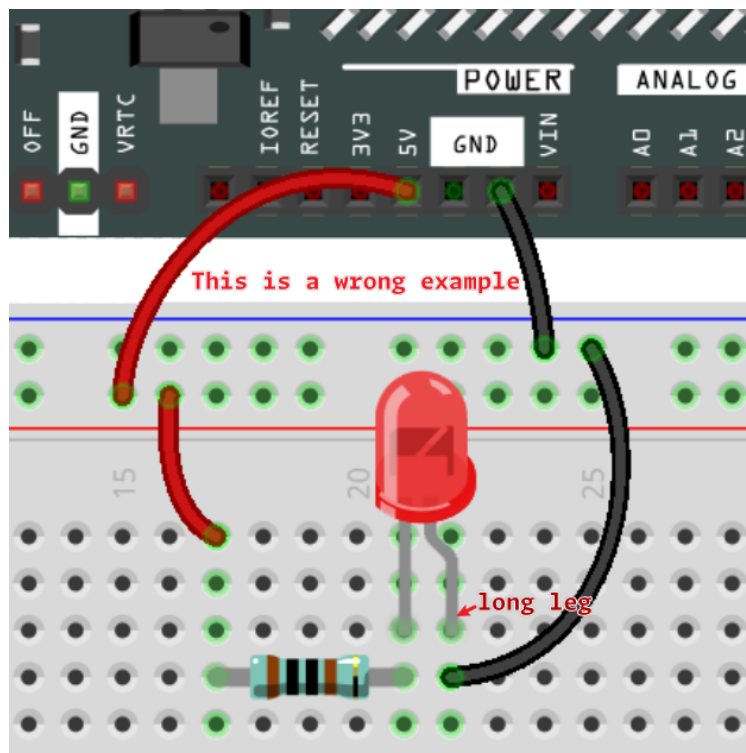
Vorsicht vor Kurzschlüssen

Kurzschlüsse können auftreten, wenn zwei Komponenten, die nicht verbunden sein sollten, „versehentlich“ verbunden werden. Dieses Kit enthält Widerstände, Transistoren, Kondensatoren, LEDs usw., die lange Metallstifte haben, die aneinanderstoßen und einen Kurzschluss verursachen können. Einige Schaltkreise funktionieren bei einem Kurzschluss einfach nicht richtig. Gelegentlich kann ein Kurzschluss Komponenten dauerhaft beschädigen, insbesondere zwischen der Stromversorgung und der Erdungsschiene, was dazu führt, dass der Schaltkreis sehr heiß wird, das Plastik auf dem Steckbrett schmilzt und sogar die Komponenten verbrennt!

Stellen Sie daher immer sicher, dass die Stifte aller Elektronik auf dem Steckbrett sich nicht berühren.

Ausrichtung des Schaltkreises

Es gibt eine Orientierung zu Schaltkreisen, und die Ausrichtung spielt bei bestimmten elektronischen Komponenten eine bedeutende Rolle. Es gibt einige Geräte mit Polarität, was bedeutet, dass sie basierend auf ihren positiven und negativen Polen korrekt angeschlossen werden müssen. Schaltkreise, die mit der falschen Ausrichtung aufgebaut werden, funktionieren nicht richtig.



Wenn Sie die LED in diesem einfachen Schaltkreis, den wir zuvor aufgebaut haben, umkehren, werden Sie feststellen, dass sie nicht mehr funktioniert.

Im Gegensatz dazu haben einige Geräte keine Richtung, wie die Widerstände in diesem Schaltkreis, sodass Sie sie umkehren können, ohne den normalen Betrieb der LEDs zu beeinflussen.

Die meisten Komponenten und Module mit Bezeichnungen wie „+“, „-“, „GND“, „VCC“ oder mit Anschlüssen unterschiedlicher Länge müssen auf eine bestimmte Weise mit dem Schaltkreis verbunden werden.

Schutz des Schaltkreises

Strom ist die Rate, mit der Elektronen an einem Punkt in einem vollständigen elektrischen Schaltkreis vorbeifließen. Im Grundsatz gilt: Strom = Fluss. Ein Ampere (AM-pir), oder Amp, ist die internationale Einheit zur Messung von Strom. Es drückt die Menge der Elektronen (manchmal auch „elektrische Ladung“ genannt) aus, die über einen bestimmten Zeitraum an einem Punkt in einem Schaltkreis vorbeifließen.

Die treibende Kraft (Spannung) hinter dem Fluss von Strom wird als Spannung bezeichnet und in Volt (V) gemessen.

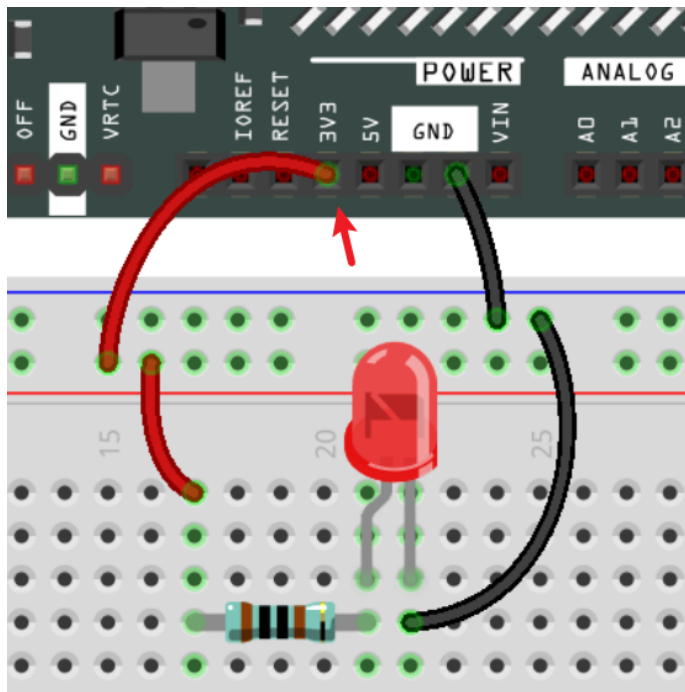
Widerstand (R) ist die Eigenschaft des Materials, die den Fluss von Strom einschränkt, und er wird in Ohm (Ω) gemessen.

Nach dem Ohmschen Gesetz (solange die Temperatur konstant bleibt), sind Strom, Spannung und Widerstand proportional. Der Strom eines Schaltkreises ist proportional zu seiner Spannung und umgekehrt proportional zu seinem Widerstand.

Daher ist Strom (I) = Spannung (V) / Widerstand (R).

- [Ohmsches Gesetz - Wikipedia](#)

Über das Ohmsche Gesetz können wir ein einfaches Experiment durchführen.



Durch das Ändern des Drahtes von 5V auf 3.3V wird die LED schwächer. Wenn Sie den Widerstand von 1000 Ohm auf 2000 Ohm ändern (Farbring: rot, schwarz, schwarz, braun und braun), werden Sie feststellen, dass die LED schwächer wird als zuvor. Je größer der Widerstand, desto schwächer die LED.

Die meisten fertig verpackten Module benötigen nur Zugang zur richtigen Spannung (normalerweise 3,3V oder 5V), wie z.B. das Ultraschallmodul.

In Ihren selbstgebaute Schaltkreise müssen Sie jedoch auf die Versorgungsspannung und die Verwendung von Widerständen für elektrische Geräte achten.

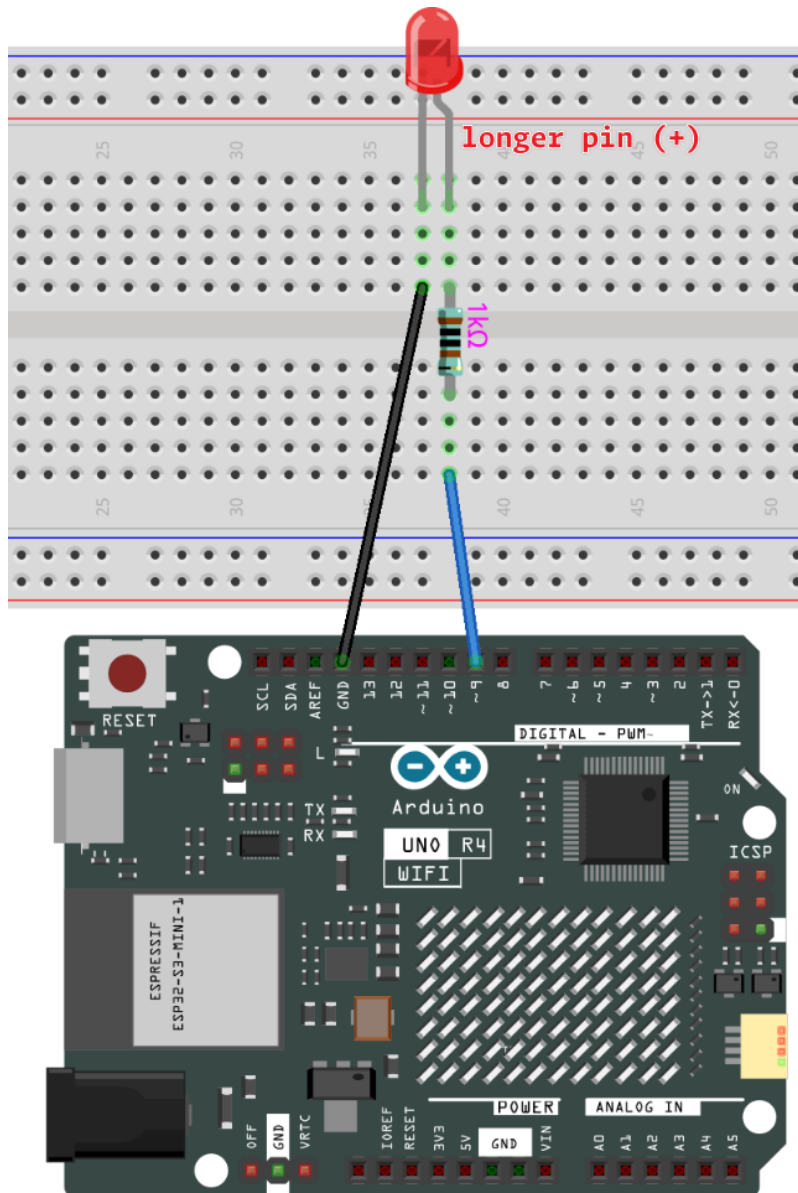
Als Beispiel verbrauchen LEDs normalerweise 20mA Strom, und ihr Spannungsabfall beträgt etwa 1,8V. Nach dem Ohmschen Gesetz benötigen wir bei einer 5V-Stromversorgung einen Widerstand von mindestens 160 Ohm $((5-1,8)/20mA)$, um zu verhindern, dass die LED durchbrennt.

Schaltkreissteuerung mit Arduino

Nachdem wir nun ein grundlegendes Verständnis von Arduino-Programmierung und elektronischen Schaltkreisen haben, ist es an der Zeit, die entscheidende Frage zu stellen: Wie steuert man Schaltkreise mit Arduino?

Einfach ausgedrückt, steuert Arduino einen Schaltkreis, indem es das Signalniveau der Pins auf dem Board ändert. Zum Beispiel, wenn eine LED auf dem Board gesteuert wird, schreibt es ein hohes oder niedriges Signalniveau auf Pin 13.

Lassen Sie uns nun versuchen, den Arduino-Board so zu programmieren, dass er die blinkende LED auf dem Steckbrett steuert. Bauen Sie den Schaltkreis so auf, dass die LED mit Pin 9 verbunden ist.



Laden Sie anschließend diesen Sketch auf das Arduino-Entwicklungsboard hoch.

```
int ledPin = 9;
int delayTime = 500;
```

(Fortsetzung auf der nächsten Seite)

```
void setup() {  
    pinMode(ledPin,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(ledPin,HIGH);  
    delay(delayTime);  
    digitalWrite(ledPin,LOW);  
    delay(delayTime);  
}
```

Dieser Sketch ist dem, den wir verwendet haben, um das Blinken der LED auf dem Board zu steuern, sehr ähnlich, der Unterschied besteht darin, dass der Wert von `ledPin` auf 9 geändert wurde. Das liegt daran, dass wir dieses Mal versuchen, das Signalniveau von Pin 9 zu steuern.

Nun können Sie sehen, wie die LED auf dem Steckbrett blinkt.

2.3.9 Wie fügt man Bibliotheken hinzu? (Wichtig)

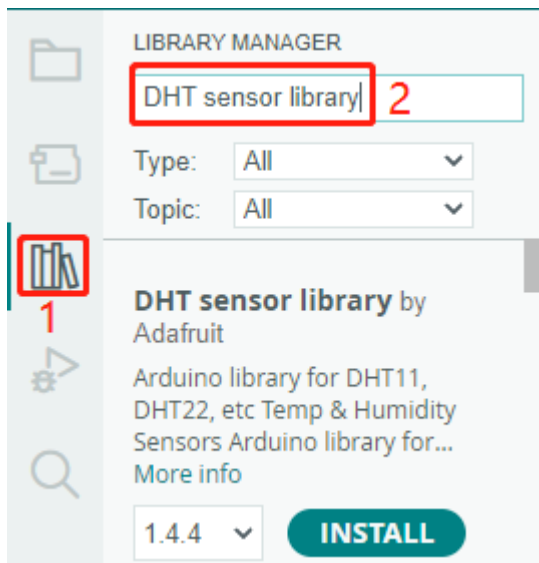
Eine Bibliothek ist eine Sammlung von vorab geschriebenem Code oder Funktionen, die die Möglichkeiten der Arduino-IDE erweitern. Bibliotheken bieten gebrauchsfertigen Code für verschiedene Funktionalitäten und ermöglichen es Ihnen, Zeit und Aufwand bei der Programmierung komplexer Funktionen zu sparen.

Verwendung des Bibliotheks-Managers

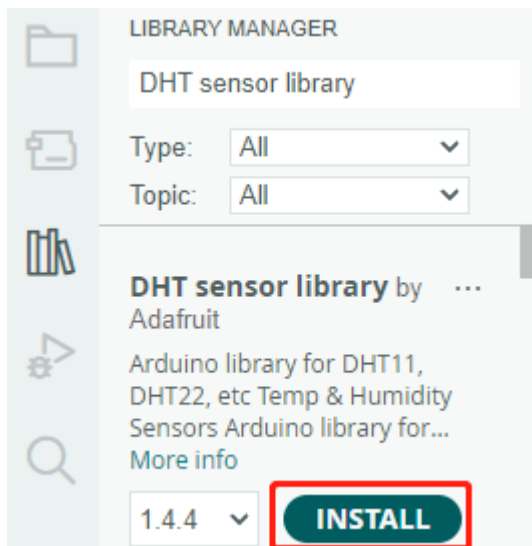
Viele Bibliotheken sind direkt über den Arduino-Bibliotheks-Manager verfügbar. Sie können den Bibliotheks-Manager wie folgt aufrufen:

1. Im **Library Manager** können Sie nach der gewünschten Bibliothek anhand ihres Namens suchen oder durch verschiedene Kategorien stöbern.

Bemerkung: In Projekten, in denen eine Bibliotheksinstallation erforderlich ist, gibt es Hinweise, welche Bibliotheken zu installieren sind. Befolgen Sie die bereitgestellten Anweisungen, wie zum Beispiel „Hier wird die DHT-Sensorbibliothek verwendet, Sie können sie aus dem Bibliotheks-Manager installieren.“ Installieren Sie einfach die empfohlenen Bibliotheken, wie vorgeschlagen.



2. Wenn Sie die gewünschte Bibliothek gefunden haben, klicken Sie darauf und dann auf den **Install**-Button.

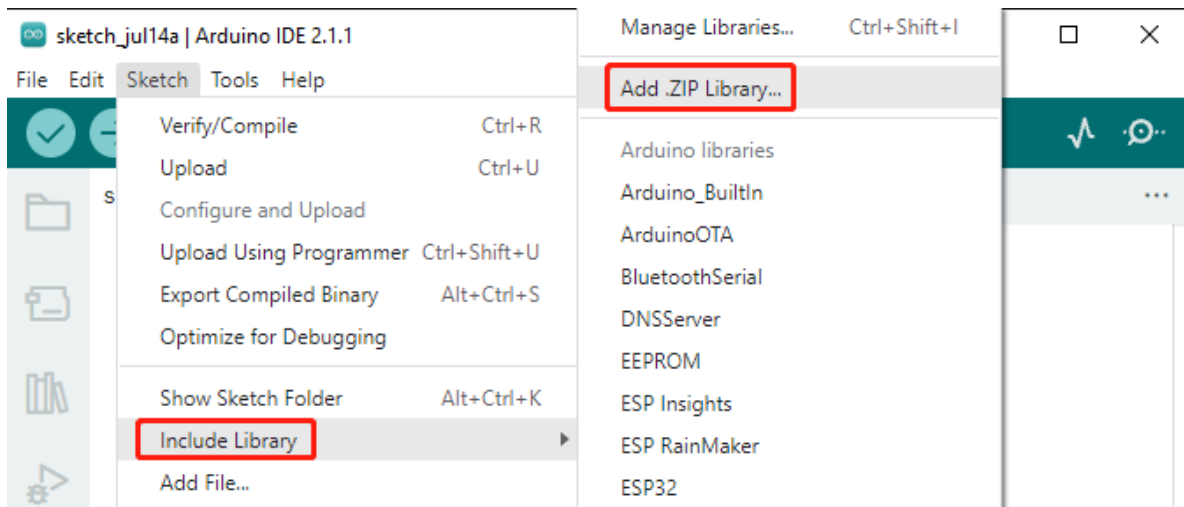


3. Die Arduino-IDE wird die Bibliothek automatisch für Sie herunterladen und installieren.

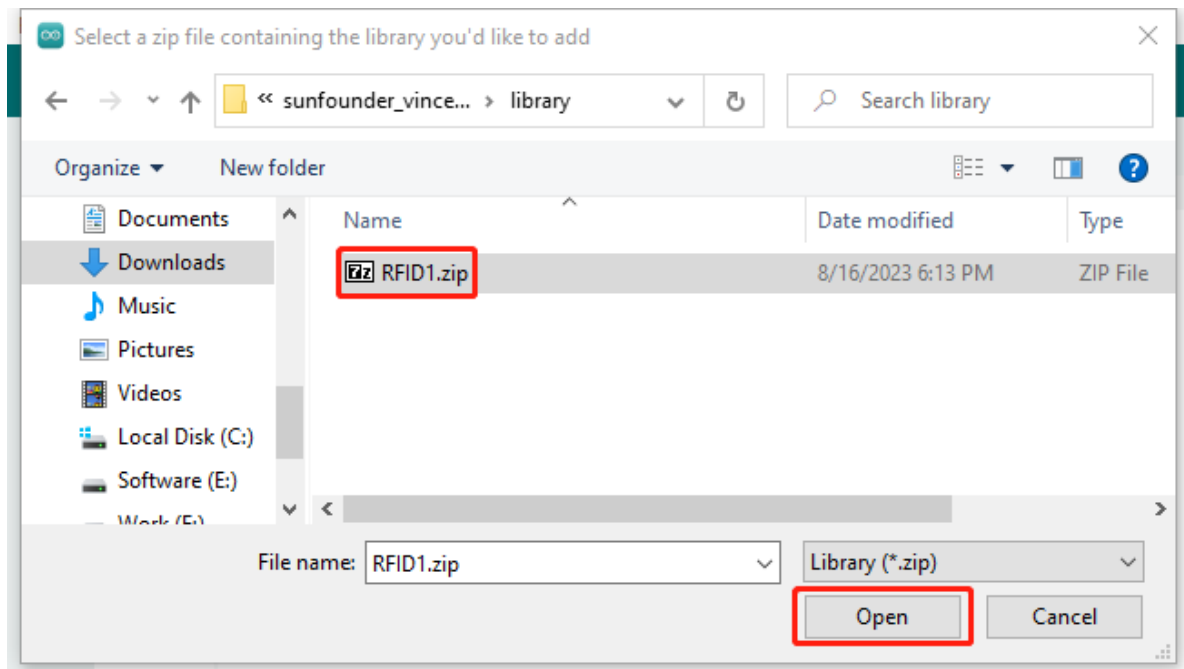
Manuelle Installation

Einige Bibliotheken sind nicht über den **Library Manager** verfügbar und müssen manuell installiert werden. Um diese Bibliotheken zu installieren, befolgen Sie diese Schritte:

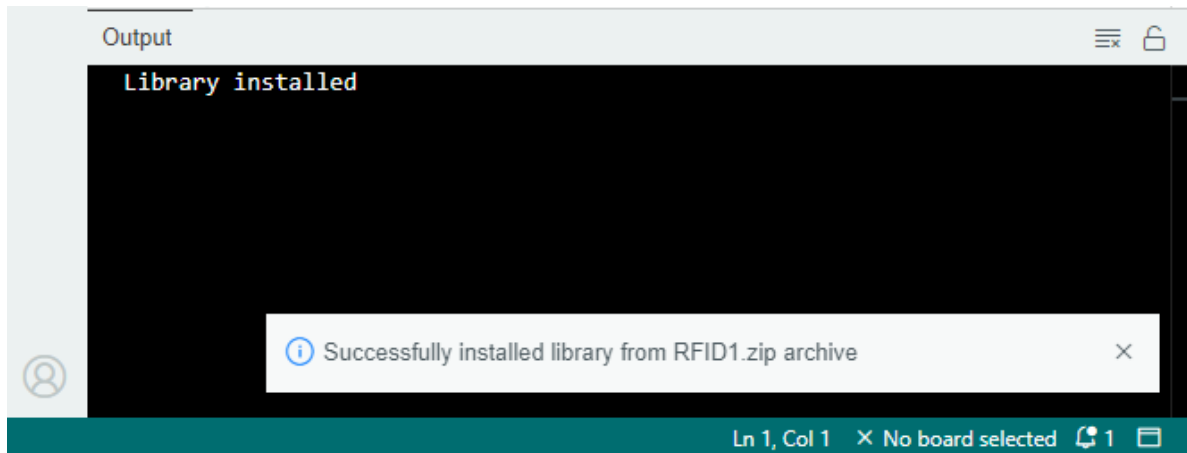
1. Öffnen Sie die Arduino-IDE und gehen Sie zu **Sketch -> Include Library -> Add .ZIP Library**.



2. Navigieren Sie zum Verzeichnis, in dem sich die Bibliotheksdateien befinden, wie zum Beispiel der Ordner `elite-explorer-kit-main/library/`, und wählen Sie die Bibliotheksdatei aus und klicken Sie auf **Open**.



3. Nach kurzer Zeit erhalten Sie eine Benachrichtigung über eine erfolgreiche Installation.

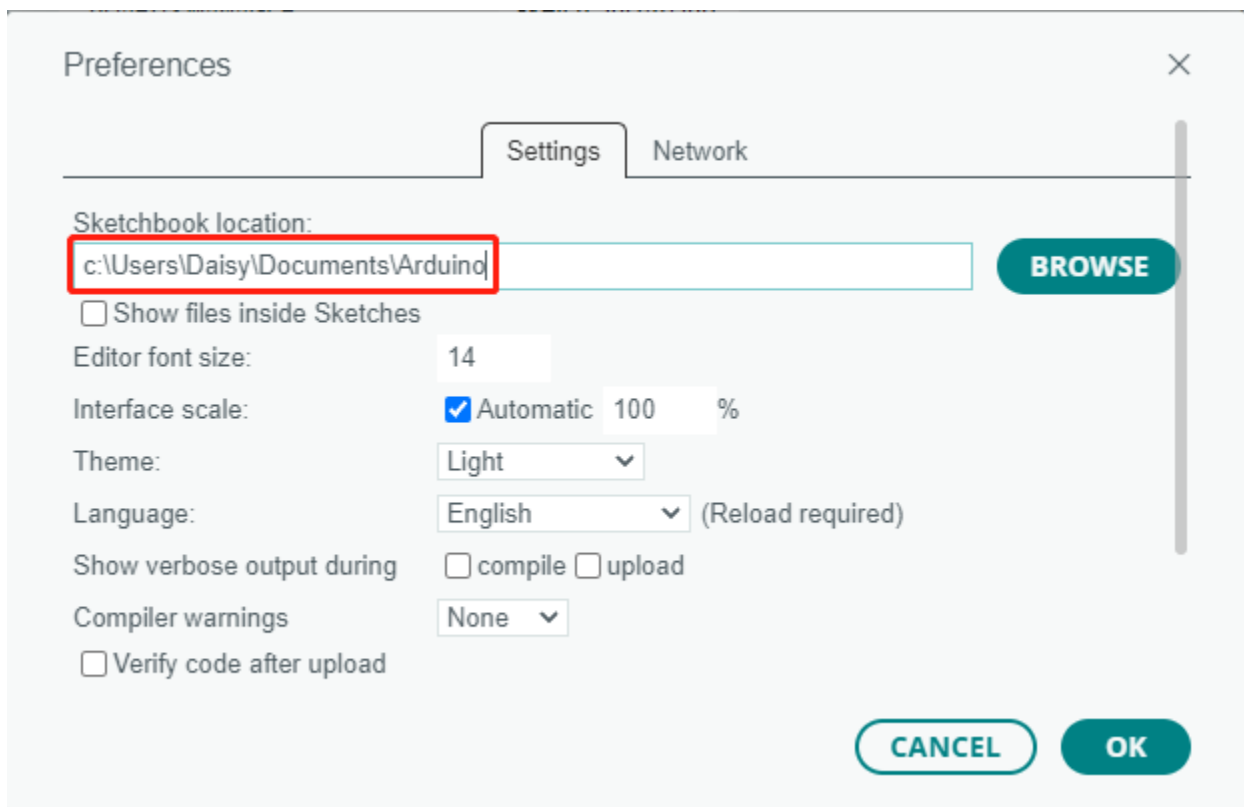


4. Wiederholen Sie den gleichen Prozess, um weitere Bibliotheken hinzuzufügen.

Bibliothekstandort

Die mit einem der oben genannten Methoden installierten Bibliotheken finden Sie im Standardbibliotheksverzeichnis der Arduino-IDE, das normalerweise unter `C:\Users\xxx\Dokumente\Arduino\libraries` liegt.

Wenn Ihr Bibliotheksverzeichnis anders ist, können Sie es überprüfen, indem Sie zu **File** -> **Preferences** gehen.



Referenz

-

KAPITEL 3

Code herunterladen

Laden Sie den relevanten Code über den untenstehenden Link herunter.

- [SunFounder Elite Explorer Kit Code](#)
- Oder sehen Sie sich den Code auf [an](#)

Grundlegende Projekte

Sensor

4.1 Fotowiderstand

4.1.1 Übersicht

In dieser Lektion lernen Sie den Fotowiderstand kennen. Fotowiderstände finden in vielen elektronischen Geräten Anwendung, wie zum Beispiel in Belichtungsmessern von Kameras, Radioweckern, Alarmanlagen (als Strahlendetektoren), kleinen Nachtlichtern, Außenuhren, Straßenlaternen mit Solarenergie und mehr. Ein Fotowiderstand in einer Straßenlampe steuert das Ein- und Ausschalten des Lichts. Umgebungslicht, das auf den Fotowiderstand fällt, bewirkt, dass Straßenlampen ein- oder ausgeschaltet werden.

4.1.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

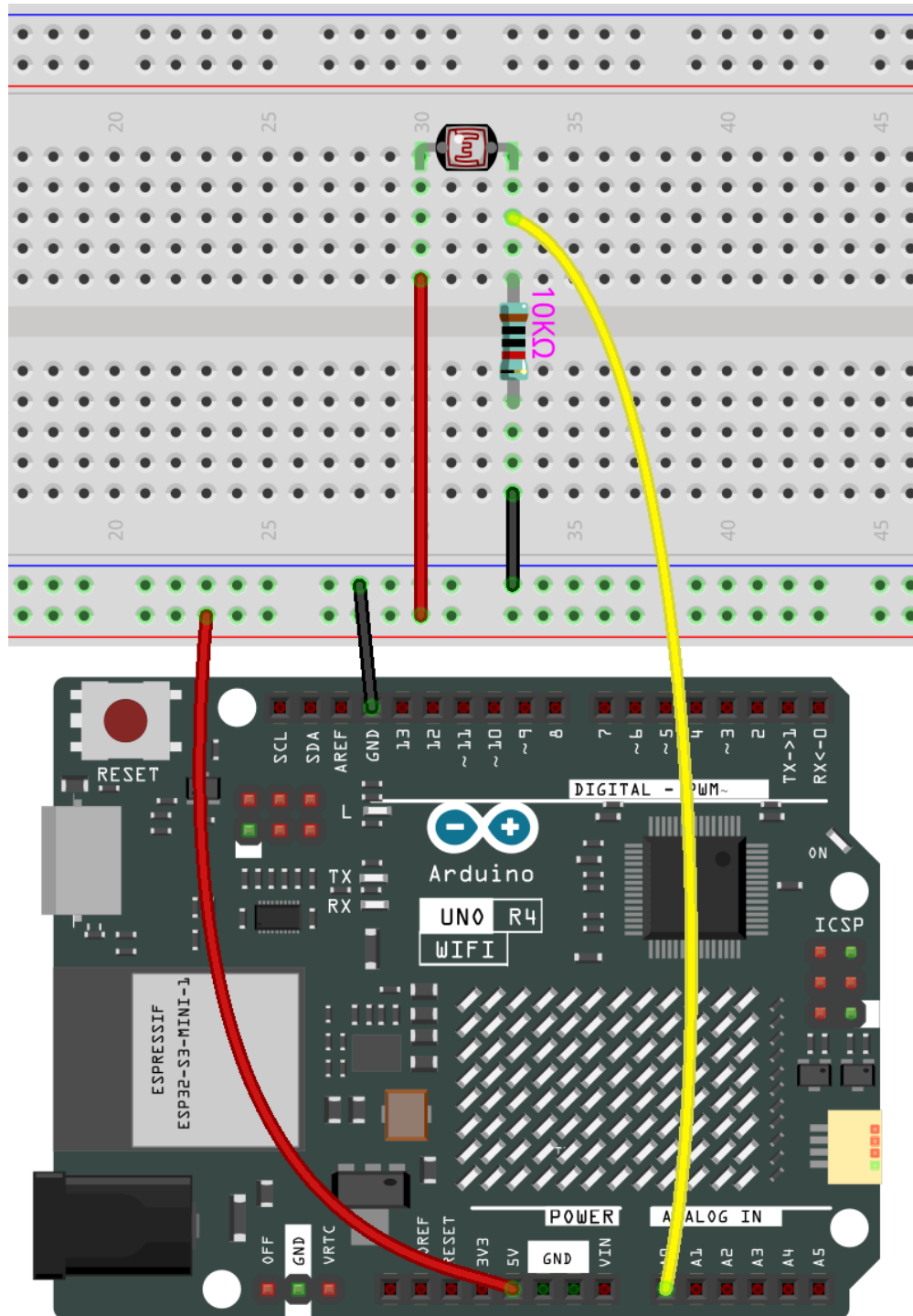
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch einzeln über die untenstehenden Links kaufen.

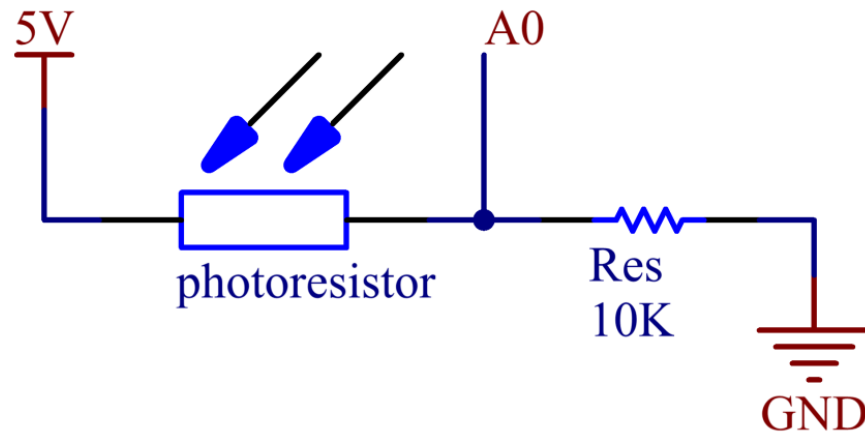
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	

4.1.3 Verdrahtung

In diesem Beispiel verwenden wir den analogen Pin (A0), um den Wert des Fotowiderstands zu lesen. Ein Pin des Fotowiderstands wird mit 5V verbunden, der andere mit A0. Außerdem wird ein 10k Widerstand benötigt, bevor der andere Pin mit GND verbunden wird.



4.1.4 Schaltplan



4.1.5 Code

Bemerkung:

- Sie können die Datei 01-fotowiderstand direkt im Pfad elite-explorer-kit-main\basic_project\01-fotowiderstand öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Nachdem Sie den Code auf das Uno-Board hochgeladen haben, können Sie den seriellen Monitor öffnen, um den ausgelesenen Wert des Pins zu sehen. Wenn das Umgebungslicht stärker wird, steigt auch der Messwert entsprechend an, und der Lesebereich des Pins liegt zwischen 0 und 1023. Jedoch kann der tatsächliche Lesebereich aufgrund der Umgebungsbedingungen und der Eigenschaften des Fotowiderstands kleiner sein als der theoretische Bereich.

4.2 Thermistor

4.2.1 Übersicht

In dieser Lektion lernen Sie, wie man einen Thermistor verwendet. Thermistoren können als elektronische Schaltkomponenten zur Temperaturkompensation in Messgeräteschaltungen eingesetzt werden. Sie finden Anwendung in Strommessgeräten, Durchflussmessern, Gasanalysatoren und anderen Geräten. Zudem werden sie für Überhitzungsschutz, kontaktlose Relais, Temperaturkonstanz, automatische Verstärkungsregelung, Motoranlauf, Zeitverzögerung, automatische Entmagnetisierung von Farbfernsehern, Brandmelder und Temperaturkompensation verwendet.

4.2.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

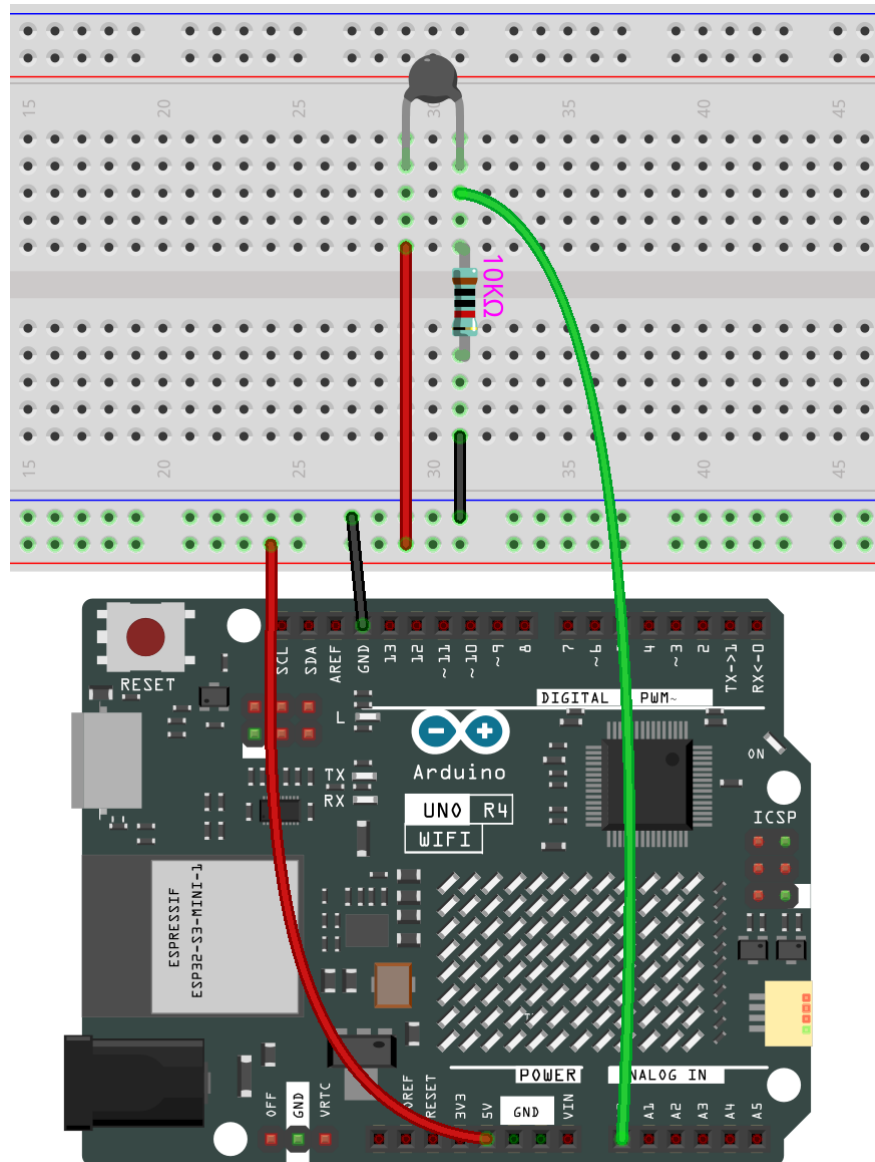
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch einzeln über die untenstehenden Links kaufen.

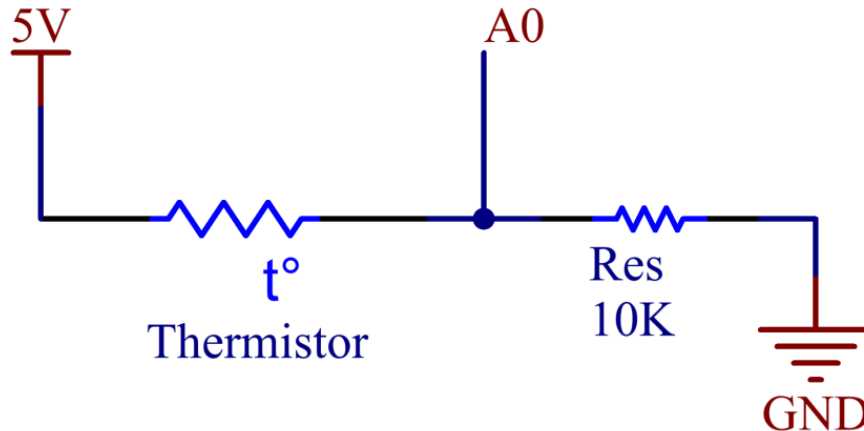
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Thermistor</i>	

4.2.3 Verdrahtung

In diesem Beispiel verwenden wir den analogen Pin A0, um den Wert des Thermistors zu ermitteln. Ein Pin des Thermistors wird mit 5V verbunden, der andere mit A0. Gleichzeitig wird ein 10k Widerstand mit dem anderen Pin verbunden, bevor dieser mit GND verbunden wird.



4.2.4 Schaltplan



4.2.5 Code

Bemerkung:

- Sie können die Datei 02-thermistor.ino direkt im Pfad elite-explorer-kit-main\basic_project\02-thermistor öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem Sie den Code auf das Uno R4 Board hochgeladen haben, können Sie den seriellen Monitor öffnen, um die aktuelle Temperatur zu überprüfen.

Die Kelvin-Temperatur wird mit der Formel $TK = 1 / (\ln(RT/RN) / B + 1 / TN)$ berechnet. Diese Gleichung stammt aus dem und vereinfacht die Berechnungen. Weitere Informationen zu dieser Formel finden Sie auf der detaillierten Einführungsseite des *Thermistor*.

4.3 Bodenfeuchtigkeit

4.3.1 Übersicht

In der Landwirtschaft können Pflanzen anorganische Elemente nicht direkt aus dem Boden aufnehmen. Stattdessen dient das im Boden vorhandene Wasser als Lösungsmittel, um diese Elemente aufzulösen.

Pflanzen absorbieren Feuchtigkeit aus dem Boden durch ihr Wurzelsystem, um Nährstoffe aufzunehmen und das Wachstum zu fördern.

Während des Wachstums und der Entwicklung von Kulturen gibt es unterschiedliche Anforderungen an die Bodentemperatur. Daher ist die Verwendung eines Bodenfeuchtigkeitssensors notwendig.

4.3.2 Benötigte Komponenten

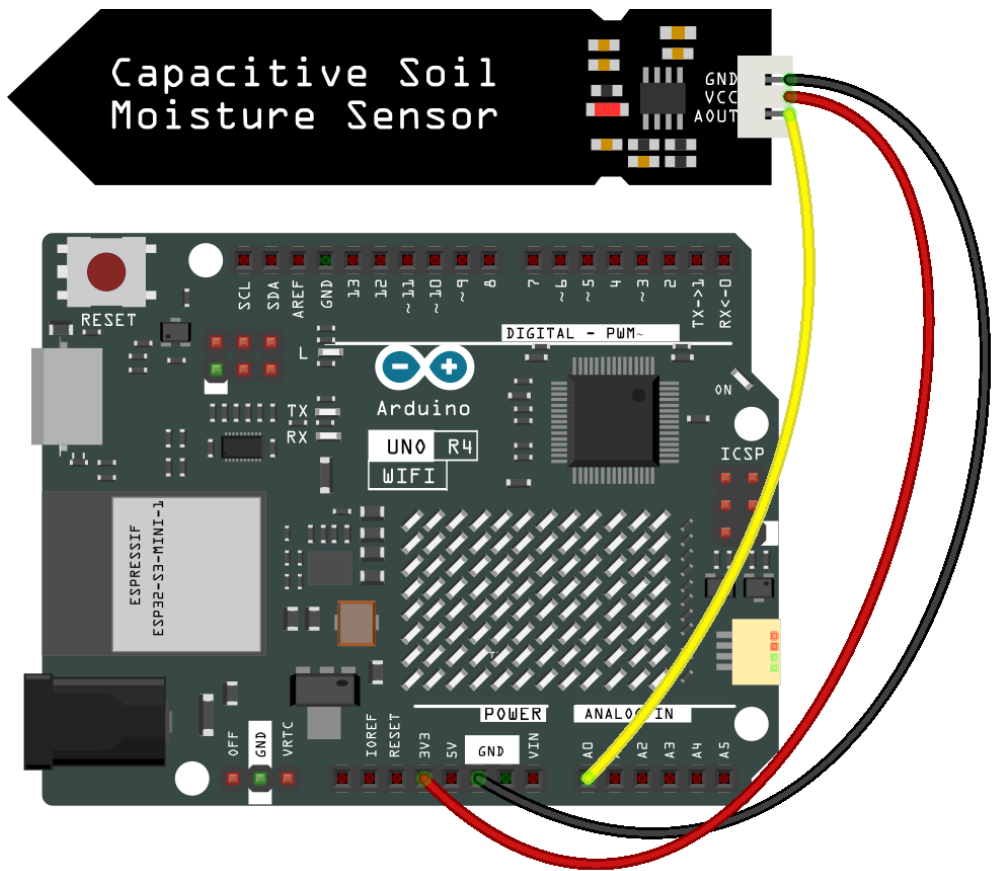
Für dieses Projekt benötigen wir die folgenden Komponenten.
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Steckbrett	
Jumperkabel	
Bodenfeuchtigkeitsmodul	

4.3.3 Verdrahtung



4.3.4 Schaltplan

4.3.5 Code

Bemerkung:

- Öffnen Sie die Datei 03-moisture.ino unter dem Pfad elite-explorer-kit-main\basic_project\03-moisture.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
-

Sobald der Code erfolgreich hochgeladen wurde, zeigt der serielle Monitor den Wert der Bodenfeuchtigkeit an.

Durch Einführen des Moduls in den Boden und Bewässerung desselben wird der Wert des Bodenfeuchtigkeitssensors kleiner.

4.4 Kippschalter

4.4.1 Übersicht

In dieser Lektion erfahren Sie mehr über den Kippschalter. Kippschalter können eingesetzt werden, um die Neigung von Objekten zu erkennen, was in der Praxis von großem Wert ist. Sie können zur Beurteilung der Neigung von Brücken, Gebäuden, Übertragungsleitungstürmen usw. verwendet werden und haben daher eine wichtige Leitfunktion bei der Durchführung von Wartungsarbeiten.

4.4.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

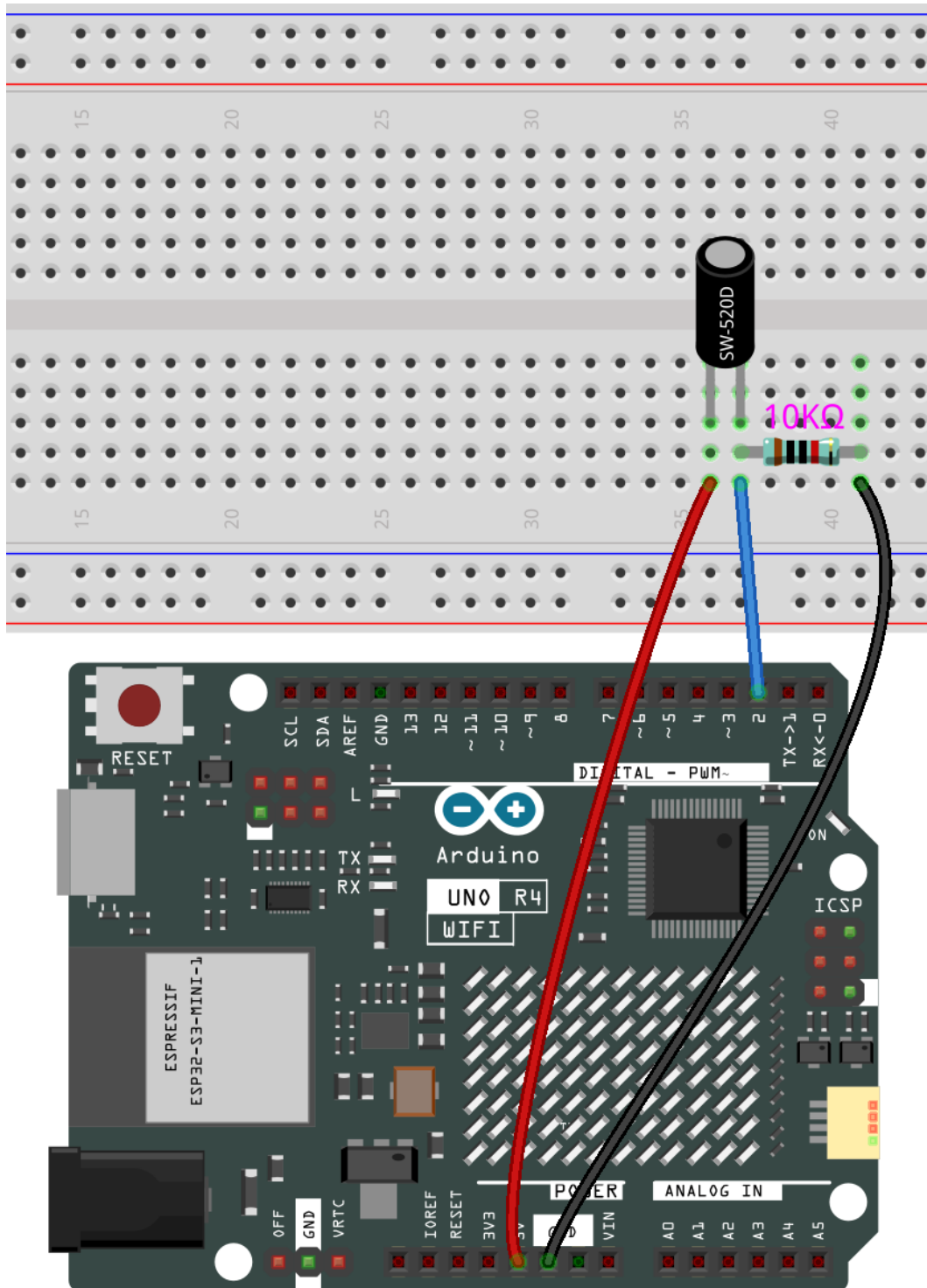
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

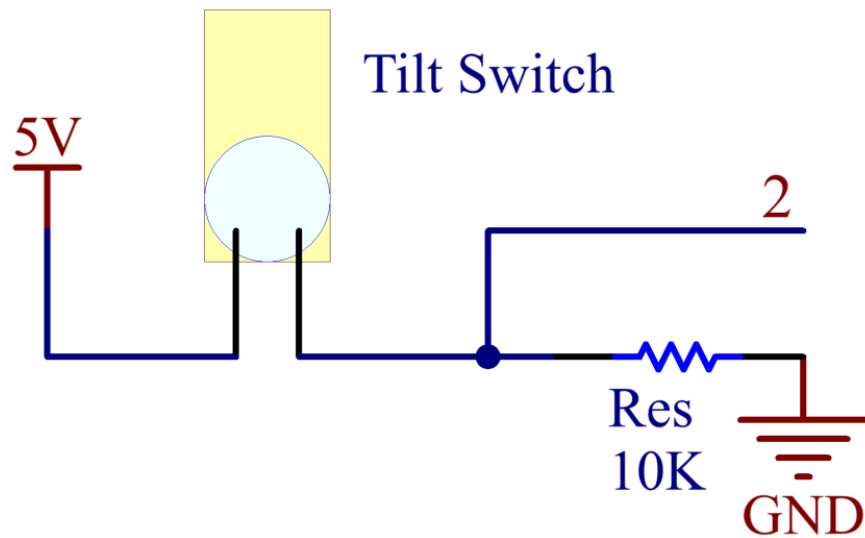
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Neigungsschalter</i>	-

4.4.3 Verdrahtung

In diesem Beispiel wird der digitale Pin 2 verwendet, um das Signal des Kippschalters zu lesen.



4.4.4 Schaltplan



4.4.5 Code

Bemerkung:

- Sie können die Datei `04-tilt_switch.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\04-tilt_switch` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Sobald die Codes auf das Uno R4-Board hochgeladen wurden, können Sie den seriellen Monitor öffnen, um die Pin-Lesungen zu betrachten. Die Lesungen zeigen entweder „1“ oder „0“ an, je nachdem, ob der Kippschalter in einer vertikalen Position ist (wobei die interne Metallkugel Kontakt mit den Drahtstiften hat) oder geneigt ist.

4.5 PIR-Bewegungssensormodul

4.5.1 Übersicht

In dieser Lektion erfahren Sie mehr über das PIR-Bewegungssensormodul. Der passive Infrarot(PIR)-Bewegungssensor ist ein Sensor, der Bewegungen erkennt. Er wird häufig in Sicherheitssystemen und automatischen Beleuchtungssystemen eingesetzt. Der Sensor hat zwei Schlitze, die Infrarotstrahlung detektieren. Wenn ein Objekt, wie eine Person, vor dem Sensor vorbeigeht, erkennt er eine Veränderung in der Menge der Infrarotstrahlung und löst ein Ausgangssignal aus.

4.5.2 Benötigte Komponenten

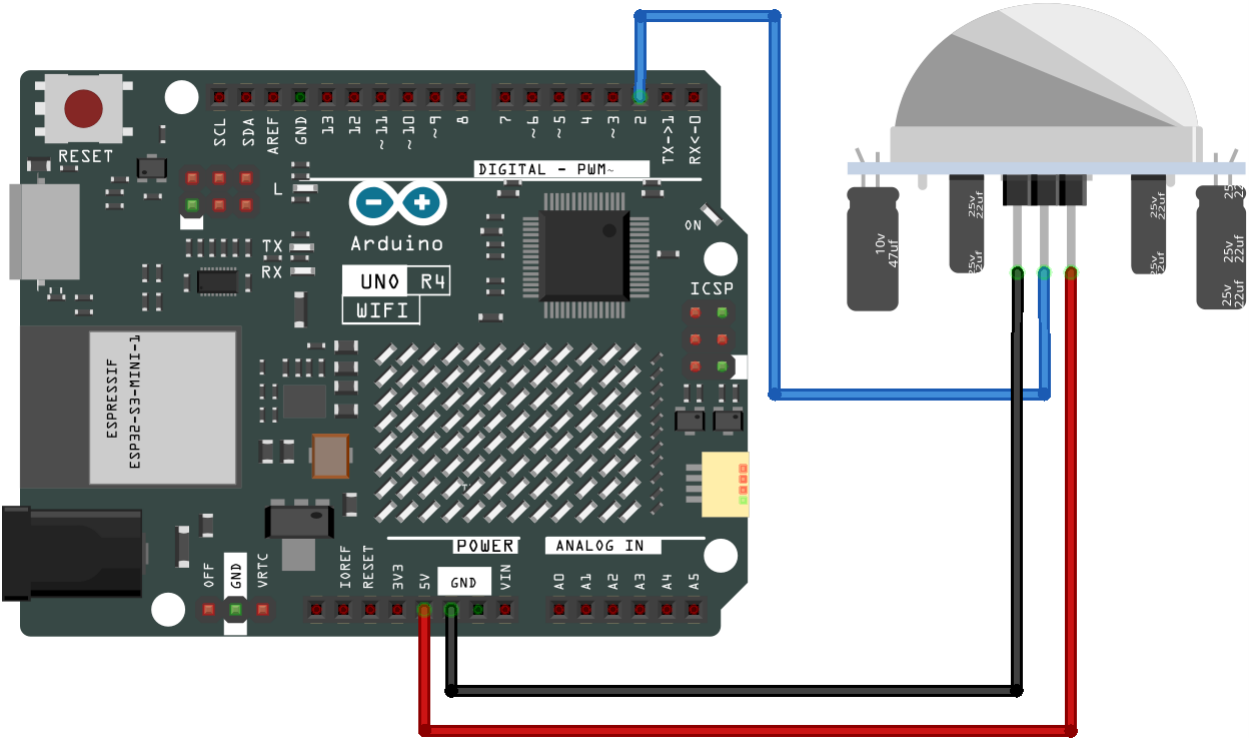
Für dieses Projekt benötigen wir die folgenden Komponenten.
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

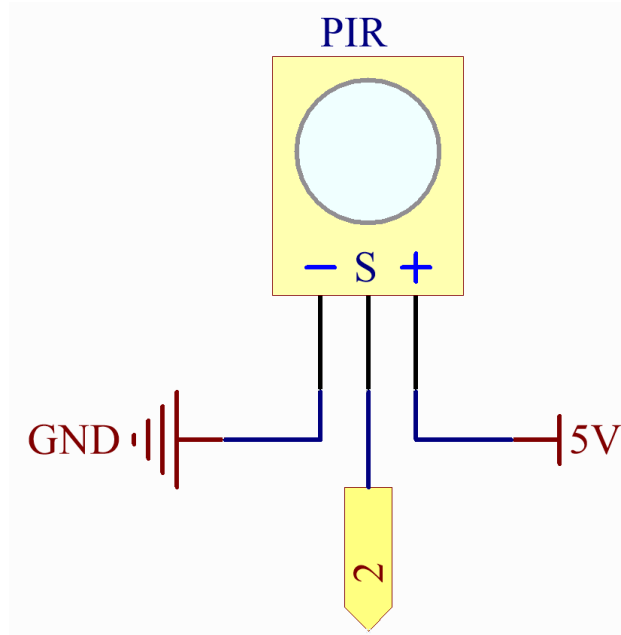
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Steckbrett	
Jumperkabel	
PIR-Bewegungssensormodul	

4.5.3 Verdrahtung



4.5.4 Schaltplan



4.5.5 Code

Bemerkung:

- Sie können die Datei 05-pir_motion_sensor.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\05-pir_motion_sensor öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem der Code auf das Arduino Uno-Board hochgeladen wurde, können Sie den seriellen Monitor öffnen, um die Ausgabe des Sensors zu beobachten. Wenn der PIR-Bewegungssensor eine Bewegung erkennt, zeigt der serielle Monitor die Nachricht „Somebody here!“ an, um anzuzeigen, dass eine Bewegung erkannt wurde. Wird keine Bewegung erkannt, wird stattdessen die Nachricht „Monitoring...“ angezeigt.

Der PIR-Sensor gibt ein digitales HIGH- oder LOW-Signal aus, je nachdem, ob Bewegung erkannt oder nicht erkannt wird. Im Gegensatz zu einem analogen Sensor, der eine Reihe von Werten liefert, ist der digitale Ausgang dieses PIR-Sensors entweder HIGH (typischerweise als „1“ dargestellt) oder LOW (typischerweise als „0“ dargestellt).

Beachten Sie, dass die tatsächliche Empfindlichkeit und der Erfassungsbereich je nach den Eigenschaften des PIR-Sensors und den Umgebungsbedingungen variieren können. Daher ist es ratsam, den Sensor entsprechend Ihren spezifischen Bedürfnissen zu kalibrieren.

4.6 Ultraschall

4.6.1 Überblick

Wenn Sie rückwärts fahren, sehen Sie die Entfernung zwischen dem Auto und den umliegenden Hindernissen, um eine Kollision zu vermeiden. Das Gerät zur Entfernungserkennung ist ein Ultraschallsensor. In diesem Experiment lernen Sie, wie die Ultraschallwelle die Entfernung erkennt.

4.6.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

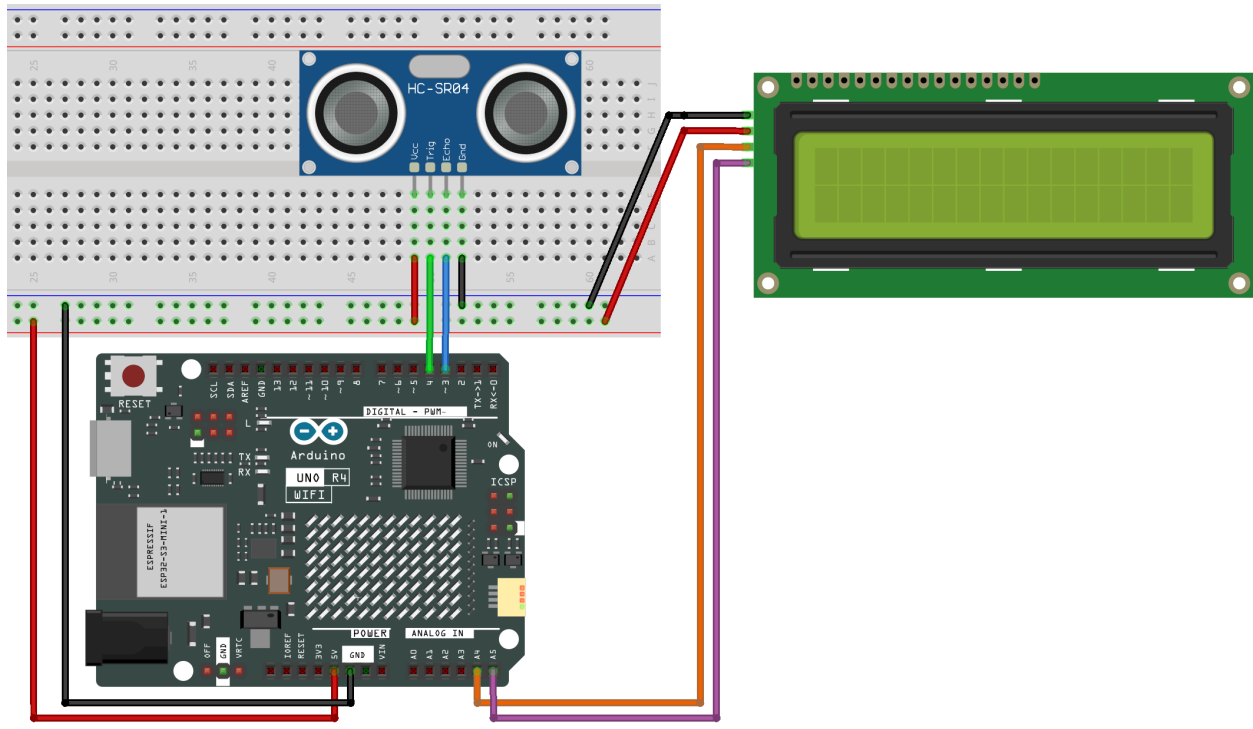
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

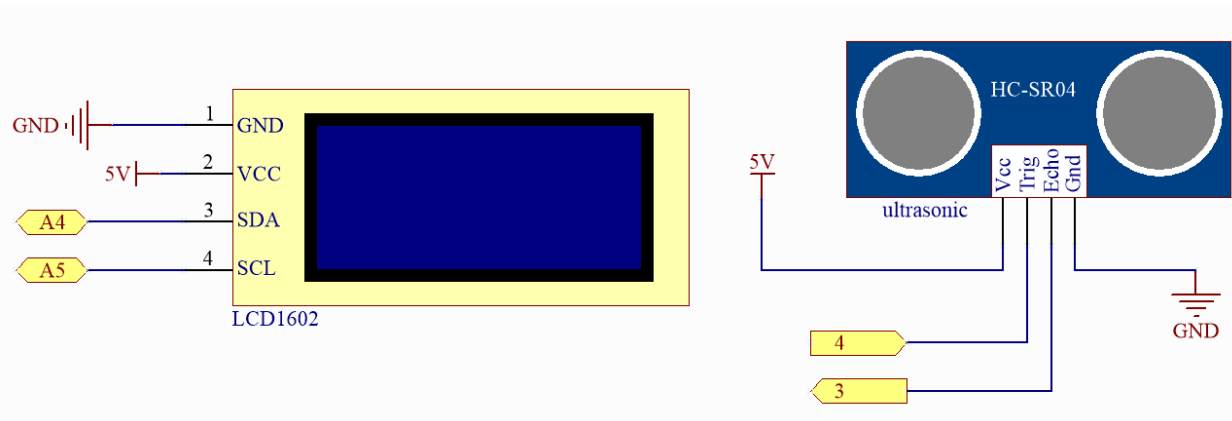
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Ultraschallmodul</i>	
<i>I2C LCD1602</i>	

4.6.3 Verdrahtung



4.6.4 Schaltplan



4.6.5 Code

Bemerkung:

- Sie können die Datei 06-ultrasonic.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\06-ultrasonic öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

4.6.6 Code-Analyse

1. Initialisieren des Ultraschallsensors und des LCD1602

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // initialize the Liquid Crystal Display
↳object with the I2C address 0x27, 16 columns and 2 rows

// Define the pin numbers for the ultrasonic sensor
const int echoPin = 3;
const int trigPin = 4;

void setup() {
  pinMode(echoPin, INPUT);           // Set echo pin as input
  pinMode(trigPin, OUTPUT);          // Set trig pin as output

  lcd.init();                        // initialize the LCD
  lcd.clear();                       // clear the LCD display
  lcd.backlight();                   // Make sure backlight is on
}
```

2. Anzeige der Entfernung auf dem LCD1602

```
void loop() {
  float distance = readDistance(); // Call the function to read the sensor
  ↳data and get the distance

  lcd.setCursor(0, 0);              //Place the cursor at Line 1, Column 1. From
  ↳here the characters are to be displayed
  lcd.print("Distance:");           //Print Distance: on the LCD
  lcd.setCursor(0, 1);              //Set the cursor at Line 1, Column 0
  lcd.print(" ");                   //Here is to leave some spaces after the
  ↳characters so as to clear the previous characters that may still remain.
  lcd.setCursor(7, 1);              //Set the cursor at Line 1, Column 7.
  lcd.print(distance);              // print on the LCD the value of the distance
  ↳converted from the time between ping sending and receiving.
  lcd.setCursor(14, 1);             //Set the cursor at Line 1, Column 14.
  lcd.print("cm");                  //print the unit "cm"

  delay(800);                       // Delay for 800 milliseconds before
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
↪repeating the loop
}
```

3. Umrechnung der Zeit in Entfernung

```
float readDistance(){// ...}
```

Hier bezieht sich „PING“ auf den Prozess, bei dem der Ultraschallsensor einen Ultraschallimpuls (oder „Ping“) aussendet und dann auf sein Echo wartet.

PING wird durch einen HIGH-Puls von 2 oder mehr Mikrosekunden ausgelöst. (Vorher einen kurzen LOW-Puls geben, um einen sauberen HIGH-Puls zu gewährleisten.)

```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

Der Echo-Pin wird verwendet, um das Signal von PING zu lesen, einen HIGH-Puls, dessen Dauer die Zeit (in Mikrosekunden) vom Senden des Pings bis zum Empfang des Echos des Objekts ist. Wir verwenden die folgende Funktion, um die Dauer zu erhalten.

```
pulseIn(echoPin, HIGH);
```

Die Schallgeschwindigkeit beträgt 340 m/s oder 29 Mikrosekunden pro Zentimeter.

Dies gibt die vom Ping zurückgelegte Strecke, hin und zurück, an, daher teilen wir durch 2, um die Entfernung des Hindernisses zu erhalten.

```
float distance = pulseIn(echoPin, HIGH) / 29.00 / 2;    // Formula: (340m/s *
↪1us) / 2
```

4.7 Feuchtigkeitssensor-Modul

4.7.1 Überblick

Feuchtigkeit und Temperatur stehen von der physikalischen Größe selbst bis hin zum tatsächlichen menschlichen Leben in enger Beziehung. Die Temperatur und Feuchtigkeit der menschlichen Umgebung beeinflussen direkt die Thermoregulationsfunktion und den Wärmeübertragungseffekt des menschlichen Körpers. Dies wirkt sich weiter auf die Denktivität und den mentalen Zustand aus und beeinflusst somit die Effizienz unseres Lernens und Arbeitens.

Die Temperatur ist eine der sieben grundlegenden physikalischen Größen im Internationalen Einheitensystem, die verwendet wird, um den Grad der Wärme oder Kälte eines Objekts zu messen. Celsius ist eine der weltweit am häufigsten verwendeten Temperaturskalen, ausgedrückt durch das Symbol „°C“.

Feuchtigkeit ist die Konzentration von Wasserdampf in der Luft. Die relative Luftfeuchtigkeit wird im Alltag häufig verwendet und in %RH ausgedrückt. Die relative Feuchtigkeit steht in engem Zusammenhang mit der Temperatur. Bei einem bestimmten Volumen abgeschlossenen Gases gilt: Je höher die Temperatur, desto niedriger die relative Feuchtigkeit und umgekehrt.

Der DHT11, ein digitaler Temperatur- und Feuchtigkeitssensor, ist in diesem Kit enthalten. Er verwendet einen kapazitiven Feuchtigkeitssensor und ein Thermistor, um die umgebende Luft zu messen und gibt ein digitales Signal am Datenpin aus.

4.7.2 Benötigte Komponenten

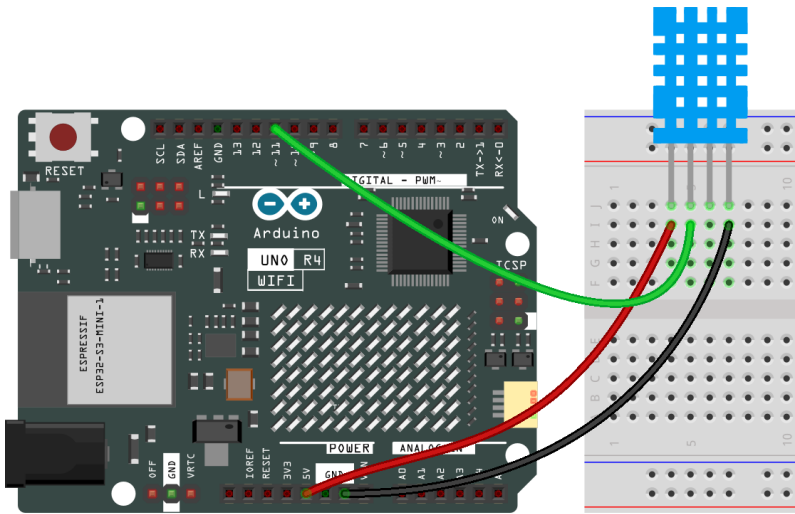
Für dieses Projekt benötigen wir die folgenden Komponenten.
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

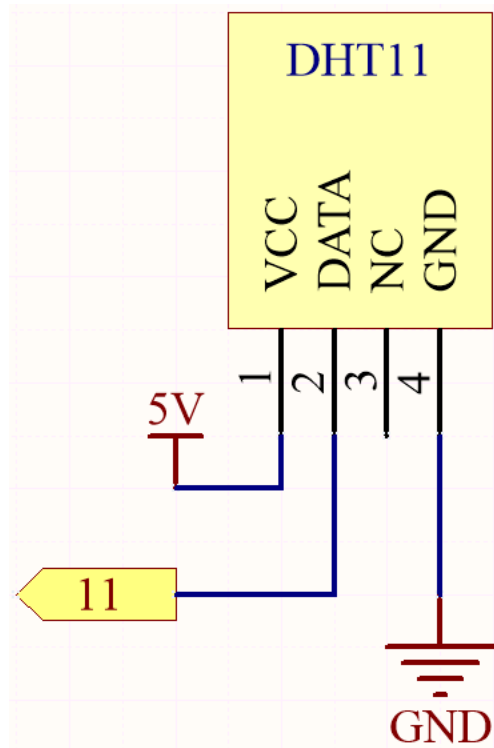
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Steckbrett	
Jumperkabel	
Feuchtigkeitssensor-Modul	

4.7.3 Verdrahtung



4.7.4 Schaltplan



4.7.5 Code

Bemerkung:

- Sie können die Datei 07-humiture_sensor.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\07-humiture_sensor öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „DHT sensor library“ und installieren Sie diese.

Nachdem der Code erfolgreich hochgeladen wurde, werden Sie sehen, dass der Serielle Monitor kontinuierlich die Temperatur und Feuchtigkeit ausgibt. Während das Programm stabil läuft, werden diese beiden Werte immer genauer.

4.7.6 Code-Analyse

1. Einbindung notwendiger Bibliotheken und Definition von Konstanten. Dieser Teil des Codes bindet die DHT-Sensorbibliothek ein und definiert die Pinnummer und den Sensortyp, die in diesem Projekt verwendet werden.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „DHT sensor library“ und installieren Sie diese.

```
#include <DHT.h>
#define DHTPIN 2          // Define the pin used to connect the sensor
#define DHTTYPE DHT11    // Define the sensor type
```

2. Erstellung eines DHT-Objekts. Hier erstellen wir ein DHT-Objekt mit der definierten Pinnummer und dem Sensortyp.

```
DHT dht(DHTPIN, DHTTYPE); // Create a DHT object
```

3. Diese Funktion wird einmal ausgeführt, wenn der Arduino startet. Wir initialisieren die serielle Kommunikation und den DHT-Sensor in dieser Funktion.

```
void setup() {
  Serial.begin(9600);
  Serial.println(F("DHT11 test!"));
  dht.begin(); // Initialize the DHT sensor
}
```

4. Hauptloop. Die Funktion loop() wird kontinuierlich nach der Setup-Funktion ausgeführt. Hier lesen wir die Feuchtigkeits- und Temperaturwerte, berechnen den Hitzeindex und drucken diese Werte auf den seriellen Monitor. Wenn das Auslesen des Sensors fehlschlägt (NaN zurückgibt), wird eine Fehlermeldung ausgegeben.

Bemerkung: Der ist eine Methode, um zu messen, wie heiß es sich draußen anfühlt, indem die Lufttemperatur und die Feuchtigkeit kombiniert werden. Es wird auch als „gefühlte Lufttemperatur“ oder „scheinbare Temperatur“ bezeichnet.

```
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("% Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

Serial.print(F("°F Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print(hif);
Serial.println(F("°F"));
}

```

4.8 RFID-RC522 Modul

4.8.1 Überblick

In dieser Lektion lernen Sie, wie man ein RFID-Modul verwendet. RFID steht für Radiofrequenz-Identifikation. Sein Funktionsprinzip beinhaltet die kontaktlose Datenkommunikation zwischen dem Lesegerät und dem Etikett zur Identifizierung des Ziels. Die Anwendungen von RFID sind umfangreich, einschließlich Tierchips, Wegfahrsperren, Zugangskontrollen, Parksteuerung, Automatisierung von Produktionsketten, Materialmanagement und vieles mehr.

4.8.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

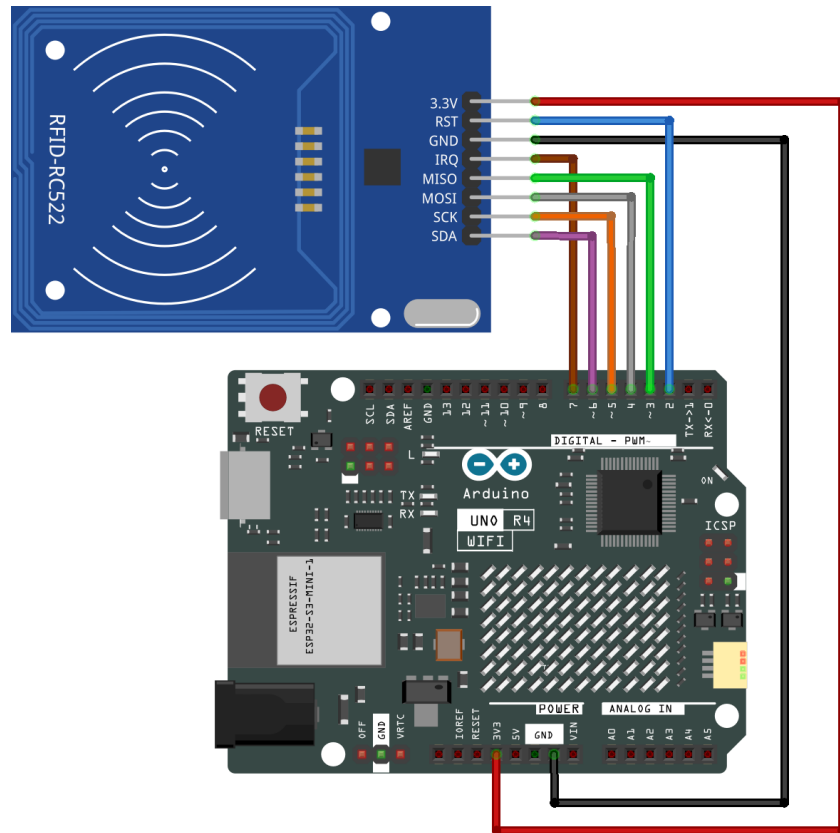
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

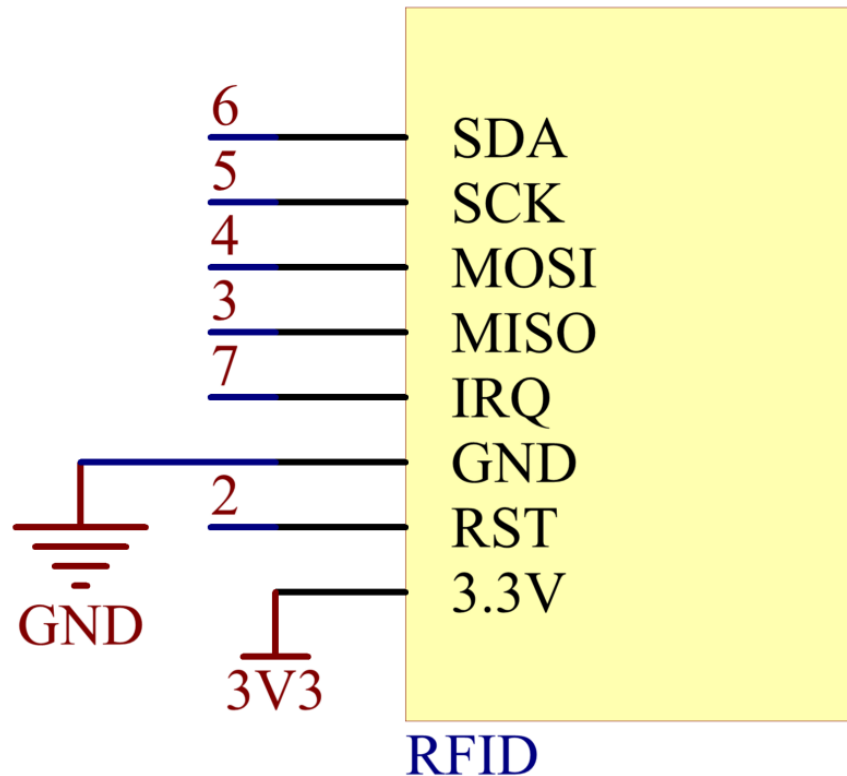
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>MFRC522-Modul</i>	

4.8.3 Verdrahtung

In diesem Beispiel stecken wir das RFID in das Breadboard. Verbinden Sie die 3.3V des RFID mit 3.3V, GND mit GND, RST mit Pin 2, SDA mit Pin 6, SCK mit Pin 5, MOSI mit Pin 4, MISO mit Pin 3 und IRQ mit Pin 7.



4.8.4 Schaltplan



4.8.5 Code

Bemerkung:

- Sie können die Datei `08-mfrc522.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\08-mfrc522` öffnen.
- Hier wird die `RFID1` Bibliothek verwendet. Die Bibliothek finden Sie im Verzeichnis `elite-explorer-kit-main/library/`, oder Sie können hier `RFID1.zip` herunterladen. Eine Anleitung zur Installation finden Sie unter [Manuelle Installation](#).

Nachdem die Codes auf das Uno-Board hochgeladen wurden, können Sie Ihre RFID-Karte (geheimer Schlüssel) in die Nähe des RFID-Lesegeräts bringen. Das Modul wird die Karteninformationen lesen und dann auf dem seriellen Monitor ausgeben.

4.8.6 Code-Analyse

Die Funktionen des Moduls sind in der Bibliothek `rfid1.h` enthalten.

```
#include <rfid1.h>
```

Bibliotheksfunktionen:

```
RFID1 rfid;
```

Erstellen einer neuen Instanz der Klasse `rfid1`, die ein bestimmtes RFID-Modul repräsentiert, welches an Ihr Arduino angeschlossen ist.

```
void begin(IRQ_PIN, SCK_PIN, MOSI_PIN, MISO_PIN, SDA_PIN, RST_PIN)
```

Pin-Konfiguration.

- `IRQ_PIN`, `SCK_PIN`, `MOSI_PIN`, `MISO_PIN`: die Pins, die für die SPI-Kommunikation verwendet werden.
- `SDA_PIN`: Synchroner Datenadapter.
- `RST_PIN`: Die Pins, die für den Reset verwendet werden.

```
void init()
```

Initialisieren des RFID.

```
uchar request(uchar reqMode, uchar *TagType);
```

Kartensuche und Kartentyp lesen, die Funktion gibt den aktuellen Lesezustand des RFID zurück und `MI_OK`, wenn erfolgreich.

- `reqMode`: Suchmethoden. `PICC_REQIDL` ist definiert als 0x26 Befehlsbits (Suchen der Karten, die nicht im Schlafmodus im Antennenbereich sind).
- `*TagType`: Wird verwendet, um den Kartentyp zu speichern, und sein Wert kann 4 Byte betragen (z.B. 0x0400).

```
char * readCardType(uchar *TagType)
```

Diese Funktion decodiert die vierstellige Hexadezimalzahl von `*tagType` in den spezifischen Kartentyp und gibt einen String zurück. Wenn 0x0400 übergeben wird, wird „MFOne-S50“ zurückgegeben.

```
uchar anticoll(uchar *serNum);
```

Konfliktvermeidung und Lesen der Karten-Seriennummer. Die Funktion gibt den aktuellen Lesestatus des RFID zurück. Sie gibt `MI_OK` zurück, wenn erfolgreich.

- `*serNum`: Wird verwendet, um die Karten-Seriennummer zu speichern, und gibt die 4-Byte-Karten-Seriennummer zurück. Das 5. Byte ist das Recheck-Byte (z.B. meine Magnetkarten-ID ist 5AE4C955).

4.9 GY-87 IMU Modul

Das GY-87 Modul ist mit drei Sensorchips ausgestattet: MPU6050, QMC5883L und BMP180, die jeweils einzigartige Fähigkeiten bieten. Der MPU6050 kombiniert ein Gyroskop und einen Beschleunigungsmesser zur Bewegungsverfolgung, der QMC5883L dient als Magnetometer zur Richtungserkennung und der BMP180 wird zur Messung des barometrischen Drucks und der Temperatur verwendet. Diese können über das I2C-Protokoll für eine effektive Kommunikation mit einem Arduino verbunden werden.

Diese Sensoren sind für eine nahtlose Integration über das I2C-Protokoll konzipiert, was eine effiziente Kommunikation mit Plattformen wie Arduino gewährleistet. Jeder Sensor im GY-87 Modul ist über einzigartige I2C-Adressen zugänglich: MPU6050 unter 0x68, QMC5883L unter 0x0D und BMP180 unter 0x77.

Einzeltutorials für jeden Sensorchip:

4.9.1 BMP180

Überblick

In diesem Tutorial beschäftigen wir uns mit dem GY-87 IMU-Modul und konzentrieren uns auf den BMP180-Sensor zur Messung von Temperatur, Druck und Höhe. Ideal für Anwendungen wie Wetterüberwachung und Höhenverfolgung behandelt diese Lektion die Verbindung des GY-87 mit einem Arduino Uno und die Verwendung der Adafruit BMP085-Bibliothek. Sie lernen, wie Sie den BMP180-Sensor initialisieren und dessen Daten auf dem Arduino-Seriellen Monitor auslesen können, eine entscheidende Fähigkeit für Projekte, die Umweltdaten benötigen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

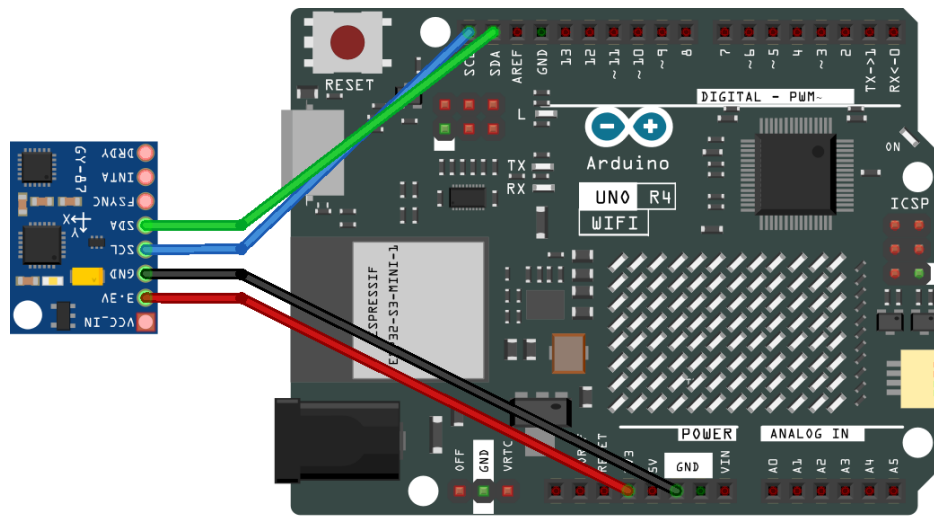
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

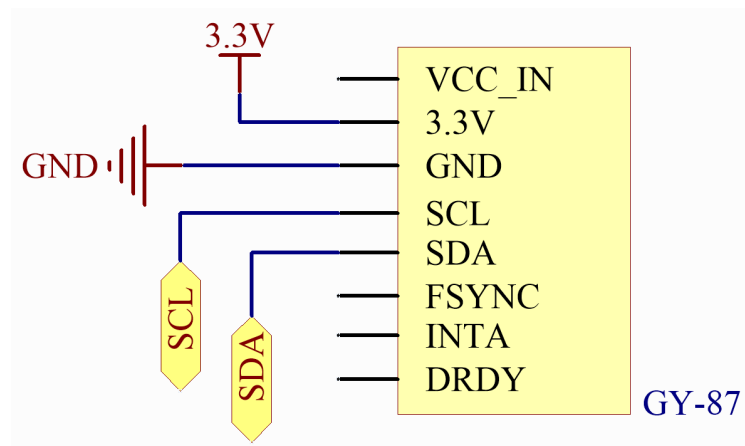
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>GY-87 IMU-Modul</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei 09-gy87_bmp180.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\09-gy87_bmp180 öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „Adafruit BMP085 Library“ und installieren Sie diese.

Code-Analyse

- initializeBMP180()

Initialisieren des BMP180-Sensors.

```
void initializeBMP180() {
  // Start BMP180 initialization
  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP180 sensor, check wiring!");
    while (1)
      ; // Halt if sensor not found
  }
  Serial.println("BMP180 Found!");
}
```

- printBMP180()

Die vom BMP180-Sensor gelesenen Werte ausdrucken.

```
void printBMP180() {
  Serial.println();
  Serial.println("BMP180 -----");
  Serial.print("Temperature = ");
  Serial.print(bmp.readTemperature());
  Serial.println(" *C");

  Serial.print("Pressure = ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pa");

  // Calculate altitude assuming 'standard' barometric
  // pressure of 1013.25 millibar = 101325 Pascal
  Serial.print("Altitude = ");
  Serial.print(bmp.readAltitude());
  Serial.println(" meters");

  Serial.print("Pressure at sealevel (calculated) = ");
  Serial.print(bmp.readSealevelPressure());
  Serial.println(" Pa");
  Serial.println("BMP180 -----");
  Serial.println();
}
```

4.9.2 MPU6050

Überblick

In diesem Tutorial lernen Sie, das GY-87 IMU-Modul mit einem Arduino Uno zu verbinden, wobei der Fokus auf dem MPU6050-Sensor liegt. Wir behandeln die Initialisierung des MPU6050 und die Anzeige seiner Beschleunigungs-, Gyroskop- und Temperaturdaten auf dem Seriellen Monitor. Diese Lektion ist unerlässlich für Projekte, die Bewegungs- und Temperatursensoren benötigen, wie Robotik, gestengesteuerte Geräte und interaktive Kunstinstallationen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

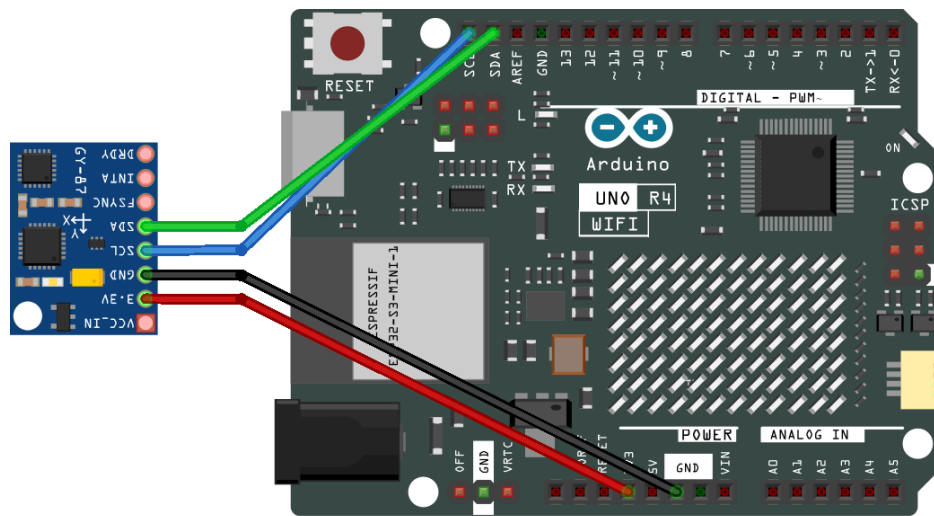
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

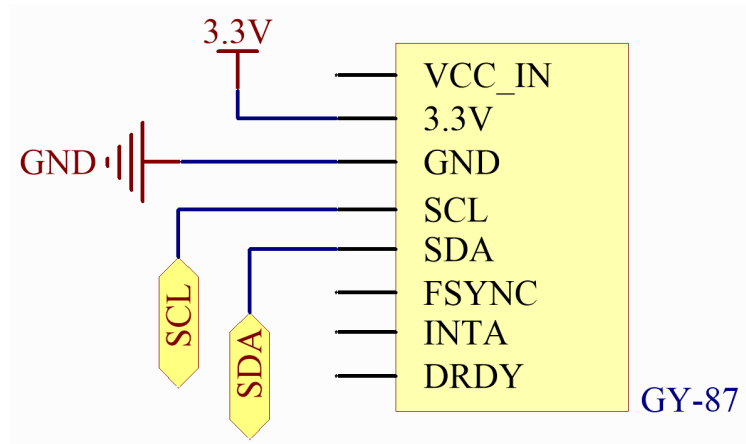
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>GY-87 IMU-Modul</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei 09-gy87_mpu6050.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\09-gy87_mpu6050 öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „Adafruit MPU6050“ und installieren Sie diese.

Code-Analyse

1. Einbinden von Bibliotheken

Die Bibliotheken Adafruit_MPU6050, Adafruit_Sensor und Wire werden für die Sensorschnittstelle und Kommunikation eingebunden.

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
```

2. Initialisieren des Sensorobjekts

Ein Objekt der Klasse Adafruit_MPU6050 wird erstellt, um den MPU6050-Sensor darzustellen.

```
Adafruit_MPU6050 mpu;
```

3. Setup-Funktion

Initialisiert die serielle Kommunikation und ruft die Funktion zur Initialisierung des MPU6050-Sensors auf.

```
void setup() {
  Serial.begin(9600);
  initializeMPU6050();
}
```

4. Loop-Funktion

Ruft wiederholt die Funktion auf, um MPU6050-Daten mit einer Verzögerung von 500 Millisekunden zwischen jedem Aufruf zu drucken.

```
void loop() {
  printMPU6050();
  delay(500);
}
```

5. Initialisieren der MPU6050-Funktion

Überprüft, ob der MPU6050 angeschlossen ist, stellt Beschleunigungsmesser- und Gyro-Bereiche ein und konfiguriert die Filterbandbreite.

```
void initializeMPU6050() {
  // Check if the MPU6050 sensor is detected
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1)
      ; // Halt if sensor not found
  }
  Serial.println("MPU6050 Found!");

  // set accelerometer range to +-8G
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);

  // set gyro range to +- 500 deg/s
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);

  // set filter bandwidth to 21 Hz
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);

  Serial.println("");
  delay(100);
}
```

6. MPU6050-Daten drucken Funktion

Liest und druckt die Beschleunigungs-, Gyroskop- und Temperaturdaten des MPU6050 auf den Seriellen Monitor.

```
void printMPU6050() {
  Serial.println();
  Serial.println("MPU6050 -----");

  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

/* Print out the values */
Serial.print("Acceleration X: ");
Serial.print(a.acceleration.x);
Serial.print(", Y: ");
Serial.print(a.acceleration.y);
Serial.print(", Z: ");
Serial.print(a.acceleration.z);
Serial.println(" m/s^2");

Serial.print("Rotation X: ");
Serial.print(g.gyro.x);
Serial.print(", Y: ");
Serial.print(g.gyro.y);
Serial.print(", Z: ");
Serial.print(g.gyro.z);
Serial.println(" rad/s");

Serial.print("Temperature: ");
Serial.print(temp.temperature);
Serial.println(" degC");

Serial.println("MPU6050 -----");
Serial.println();
}

```

4.9.3 QMC5883L

Überblick

In diesem Tutorial befassen wir uns mit dem GY-87 IMU-Modul und konzentrieren uns auf das QMC5883L-Magnetometer. Der erste Teil des Tutorials führt Sie durch die Kalibrierung des QMC5883L-Magnetometers, was für genaue Messungen des Magnetfelds unerlässlich ist. Sie lernen, wie man einen Kalibrierungssketch auf Arduino hochlädt, eine Echtzeit-Kalibrierung durchführt und diese Einstellungen in Ihren Projekten anwendet. Der zweite Teil des Tutorials behandelt die Initialisierung des MPU6050 (Beschleunigungsmesser und Gyroskop) und QMC5883L auf einem Arduino Uno unter Verwendung der Adafruit MPU6050- und QMC5883LCompass-Bibliotheken. Sie erfahren, wie man Sensordaten liest und auf dem Seriellen Monitor anzeigt, was eine grundlegende Fähigkeit für Anwendungen in Navigation, Bewegungstracking und Orientierungserkennung ist.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

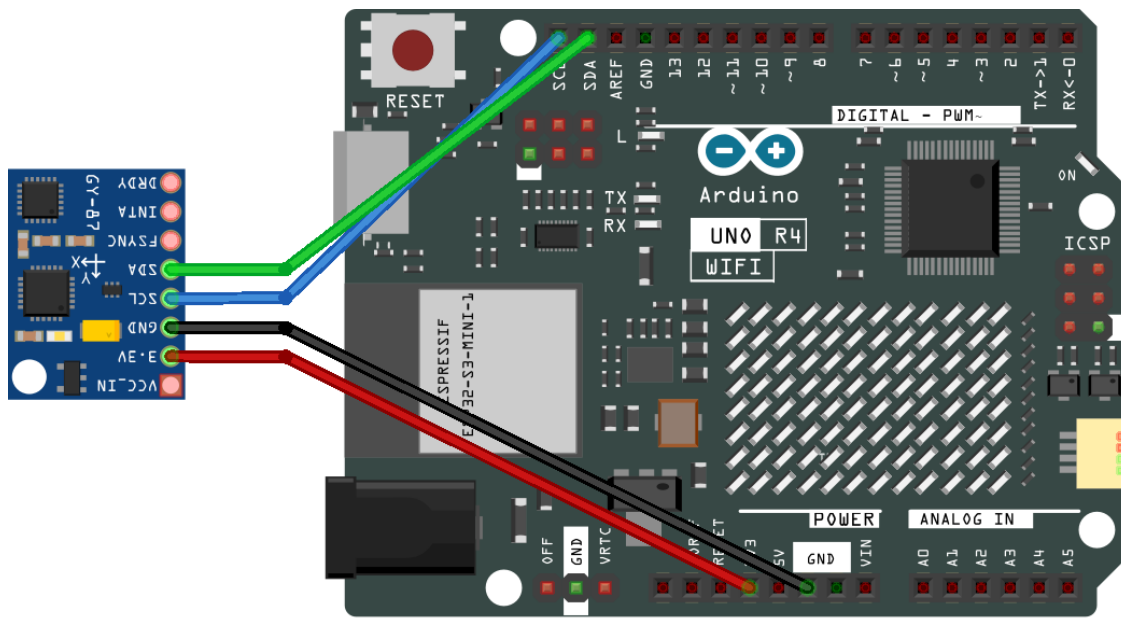
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

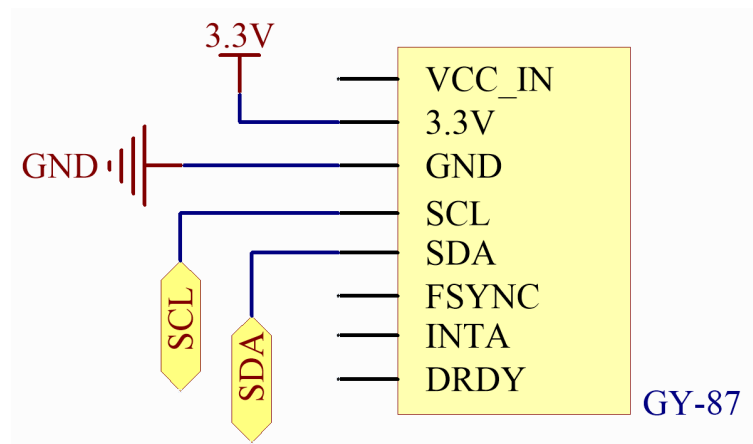
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Jumperkabel	
GY-87 IMU-Modul	-

Verdrahtung



Schaltplan

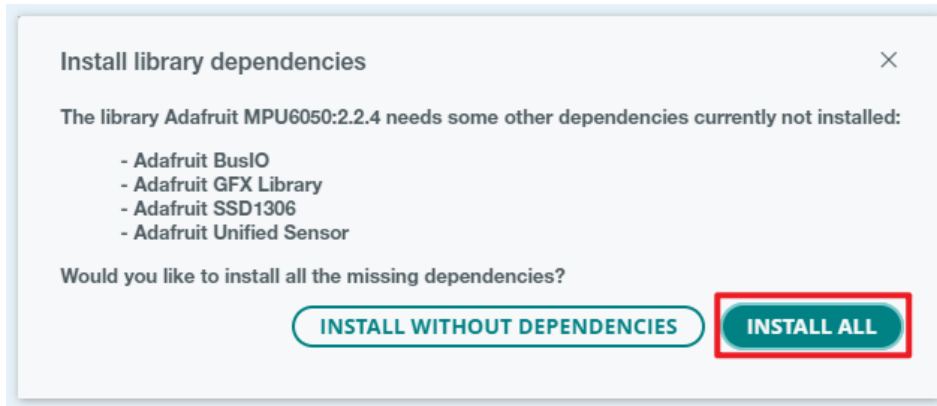


Bibliothek installieren

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager.

- Suchen Sie nach „**Adafruit MPU6050**“ und installieren Sie

Bei der Installation jeder Bibliothek stellen Sie bitte sicher, dass die Installation aller Abhängigkeiten ausgewählt wird.



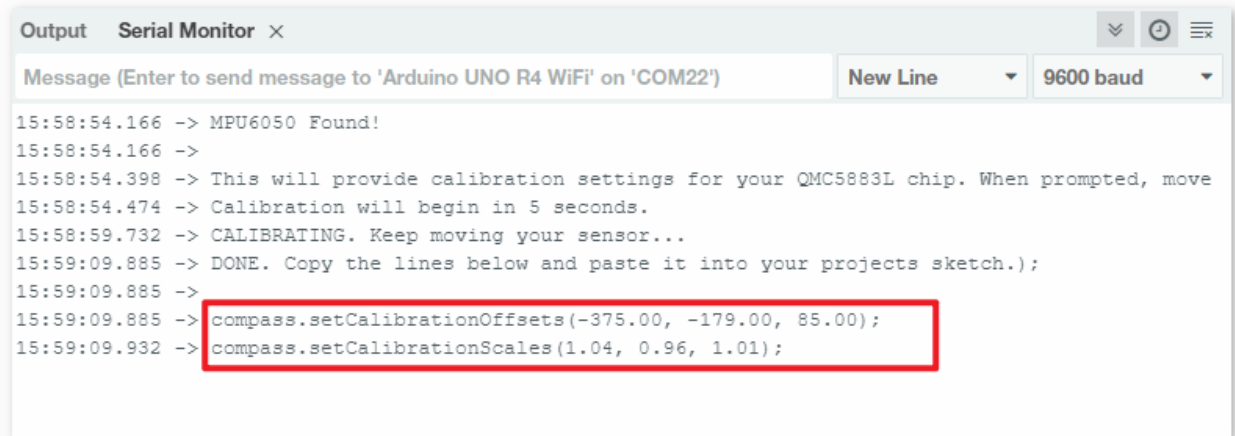
- Suchen Sie nach „**QMC5883LCompass**“ und installieren Sie

QMC5883L kalibrieren

Bemerkung:

- Sie können die Datei `09-gy87_compass_calibration.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\09-gy87_compass_calibration` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem Sie den Code hochgeladen haben, öffnen Sie den seriellen Monitor. Befolgen Sie die Anweisungen im seriellen Monitor, um QMC5883L zu kalibrieren. Wenn Sie aufgefordert werden, den Sensor zu bewegen, wird die Kalibrierungsmethode „Achterfigur“ empfohlen. Alternativ können Sie den Sensor einfach parallel zum Boden halten und ihn im Uhrzeigersinn oder gegen den Uhrzeigersinn drehen, bis der serielle Monitor meldet, dass die Kalibrierung abgeschlossen ist.



Sobald alle Kalibrierungsdaten gesammelt wurden, wird das Sketch Ihnen einige Codezeilen geben, die wie `compass.setCalibrationOffsets(-375.00, -179.00, 85.00);` und `compass.setCalibrationScales(1.04, 0.96, 1.01);` aussehen. Kopieren Sie diesen Code. Es ist empfehlenswert, ihn für zukünftige Referenzen zu speichern.

Bei Verwendung von QMC5883L: Öffnen Sie den Sketch Ihres Projekts und fügen Sie die kopierte Codezeile direkt unter den Aufruf `compass.init()` ein. So wie hier:

```
void initializeQMC5883L() {  
  
    compass.init();  
  
    // You should replace the code below according to your calibration results  
    compass.setCalibrationOffsets(-375.00, -179.00, 85.00);  
    compass.setCalibrationScales(1.04, 0.96, 1.01);  
  
}
```

Code

Bemerkung: Magnetometer müssen kalibriert(*QMC5883L kalibrieren*) werden, bevor sie als Kompass verwendet werden können, und müssen waagrecht gehalten und **fern von Eisenobjekten, magnetisierten Materialien und stromführenden Drähten** gehalten werden.

Bemerkung:

- Sie können die Datei `09-gy87_qmc5883l.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\09-gy87_qmc5883l` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.
- Fügen Sie den Code, den Sie aus den Kalibrierungsschritten erhalten haben, unter die Zeile `compass.init()` in der Funktion `initializeQMC5883L()` ein.

Code-Analyse

1. Einbinden von Bibliotheken und Initialisieren von Sensoren Dieser Abschnitt enthält die notwendigen Bibliotheken für die MPU6050- und QMC5883L-Sensoren und initialisiert ihre Objekte.

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <QMC5883LCompass.h>

Adafruit_MPU6050 mpu;
QMC5883LCompass compass;
```

2. Setup-Funktion

Initialisiert die serielle Kommunikation, den MPU6050-Sensor und setzt den MPU6050 in den I2C-Bypass-Modus, um direkten Zugriff auf das QMC5883L-Magnetometer zu ermöglichen. Anschließend wird das QMC5883L-Magnetometer initialisiert.

```
void setup() {
  // Initialize the serial communication with a baud rate of 9600
  Serial.begin(9600);

  // Initialize the MPU6050 sensor (accelerometer and gyroscope)
  initializeMPU6050();

  // Enable I2C bypass on MPU6050 to directly access the QMC5883L magnetometer
  mpu.setI2CBypass(true);

  // Initialize the QMC5883L magnetometer sensor
  initializeQMC5883L();
}
```

3. Loop-Funktion

Liest kontinuierlich Daten vom QMC5883L-Magnetometer und gibt sie auf dem Seriellen Monitor aus.

```
void loop() {
  printQMC5883L();
  delay(500);
}
```

4. Initialisieren der QMC5883L-Funktion

Initialisiert und kalibriert das QMC5883L-Magnetometer. Die Kalibrierungswerte sollten auf Grundlage spezifischer Kalibrierungsdaten angepasst werden. (*QMC5883L kalibrieren*)

```
void initializeQMC5883L() {
  compass.init();

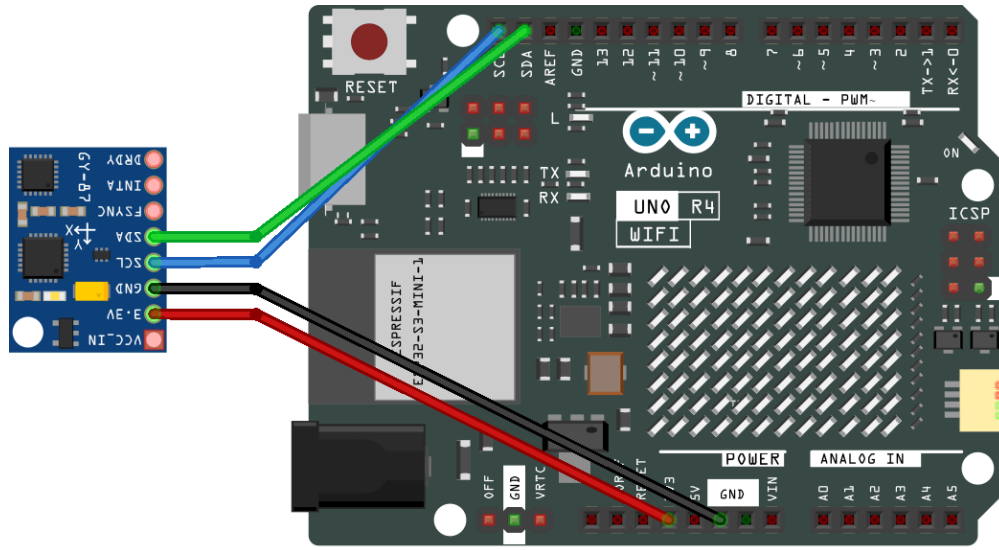
  // You should replace the code below according to your calibration results
  compass.setCalibrationOffsets(-549.00, -66.00, 160.00);
  compass.setCalibrationScales(0.97, 1.02, 1.02);
}
```

5. QMC5883L-Daten drucken Funktion

Diese Funktion liest die X-, Y-, Z-Werte und den Azimut des Magnetometers und druckt sie auf den Seriellen Monitor.

```
void printQMC5883L() {  
  
    Serial.println();  
    Serial.println("QMC5883L -----");  
  
    int x, y, z, a;  
    char myArray[3];  
  
    compass.read();  
  
    x = compass.getX();  
    y = compass.getY();  
    z = compass.getZ();  
  
    a = compass.getAzimuth();  
  
    compass.getDirection(myArray, a);  
  
    Serial.print("X: ");  
    Serial.print(x);  
  
    Serial.print(" Y: ");  
    Serial.print(y);  
  
    Serial.print(" Z: ");  
    Serial.print(z);  
  
    Serial.print(" Azimuth: ");  
    Serial.print(a);  
  
    Serial.print(" Direction: ");  
    Serial.print(myArray[0]);  
    Serial.print(myArray[1]);  
    Serial.println(myArray[2]);  
  
    Serial.println("QMC5883L -----");  
    Serial.println();  
}
```

Wenn Sie diese drei Chips gleichzeitig verwenden möchten, finden Sie hier ein einfaches Beispiel:

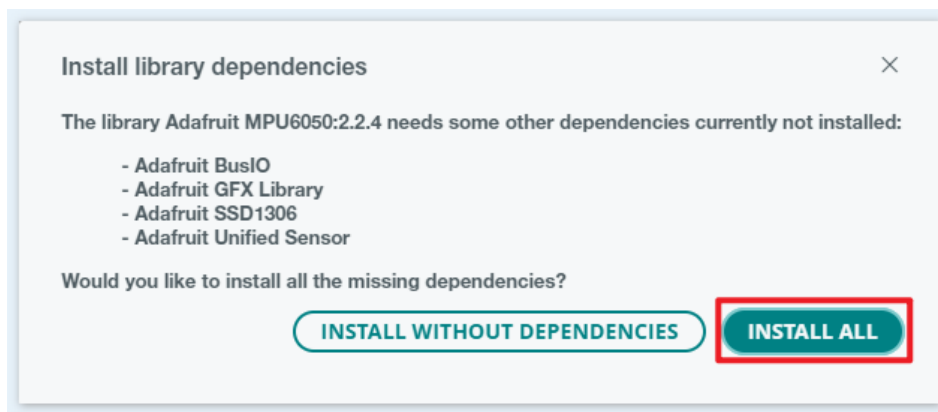
**Bemerkung:**

- Sie können die Datei 09-gy87.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\09-gy87 öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager.

- Suchen Sie nach „**Adafruit MPU6050**“ und installieren Sie

Bei der Installation jeder Bibliothek stellen Sie bitte sicher, dass die Installation aller Abhängigkeiten ausgewählt wird.



- Suchen Sie nach „**Adafruit Unified Sensor**“ und installieren Sie
- Suchen Sie nach „**QMC5883LCompass**“ und installieren Sie
- Suchen Sie nach „**Adafruit BMP085 Library**“ und installieren Sie

Bemerkung: Magnetometer müssen kalibriert werden ([QMC5883L kalibrieren](#)), bevor sie als Kompass verwendet werden können, und müssen waagrecht gehalten und **fern von Eisenobjekten, magnetisierten Materialien und**

stromführenden Drähten gehalten werden.

Nachdem der Code erfolgreich auf Ihrem Arduino Uno R4 hochgeladen wurde, wird der Serielle Monitor aktiv und druckt kontinuierlich Sensordaten vom GY-87 IMU-Modul aus. Dieses Modul beinhaltet drei einzelne Sensoren: den MPU6050 für Beschleunigungs- und Gyroskopdaten, den QMC5883L für Magnetometerdaten und den BMP180 für barometrische Druck- und Temperaturdaten.

Anzeige

4.10 LED-Modul

Genau wie das Drucken von „Hallo, Welt!“ der erste Schritt beim Erlernen der Programmierung ist, so ist das Ansteuern einer LED die traditionelle Einführung in die physische Programmierung.

4.10.1 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

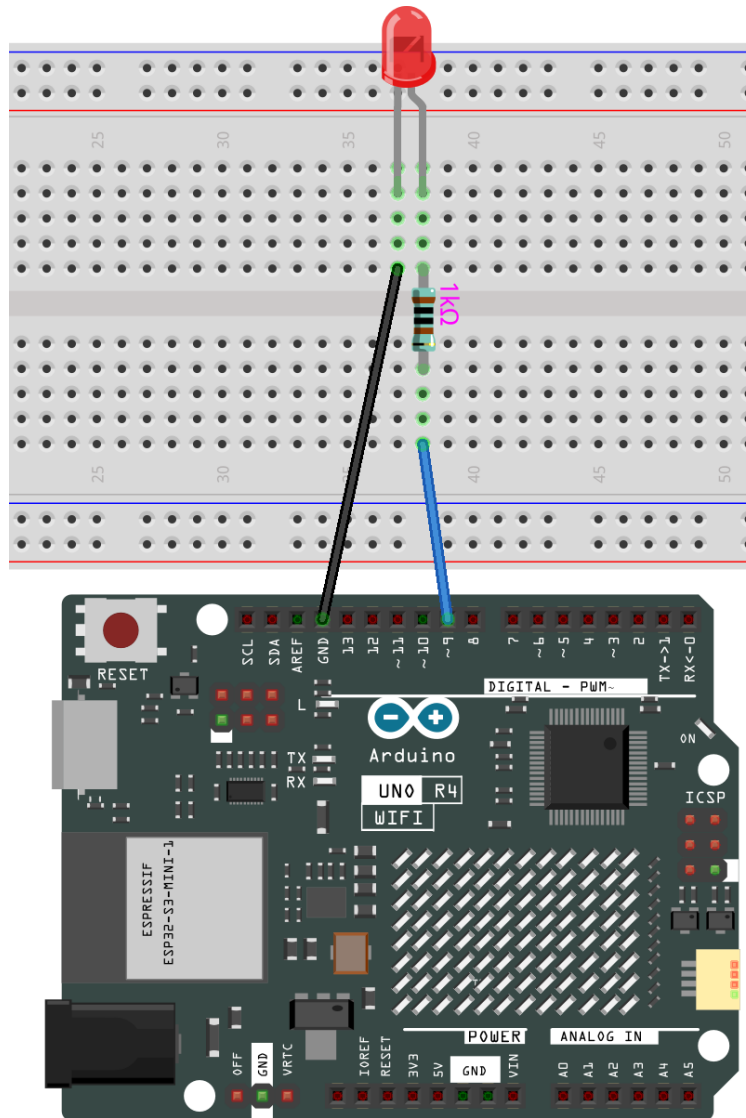
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

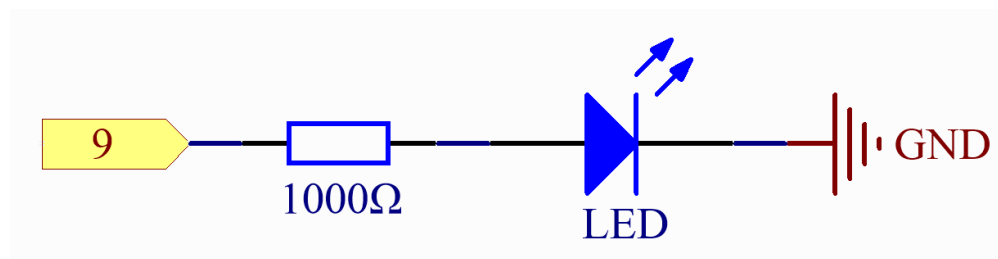
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

4.10.2 Verdrahtung



4.10.3 Schaltplan



4.10.4 Code

Bemerkung:

- Sie können die Datei 10-led.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\10-led öffnen.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
-

Nachdem der Code erfolgreich hochgeladen wurde, sehen Sie, wie die an den digitalen Pin 9 des Arduino-Boards angeschlossene LED zu blinken beginnt. Die LED leuchtet eine halbe Sekunde lang auf und schaltet sich dann für eine weitere halbe Sekunde aus, wobei dieser Zyklus kontinuierlich wiederholt wird, solange das Programm läuft.

4.10.5 Code-Analyse

Hier verbinden wir die LED mit dem digitalen Pin 9, daher müssen wir eine `int` Variable namens `ledPin` zu Beginn des Programms deklarieren und den Wert 9 zuweisen.

```
const int ledPin = 9;
```

Jetzt initialisieren wir den Pin in der `setup()`-Funktion, wo Sie den Pin auf `OUTPUT`-Modus einstellen müssen.

```
void setup() {  
    pinMode(ledPin, OUTPUT);  
}
```

In `loop()`, wird `digitalWrite()` verwendet, um dem `ledPin` ein 5V-High-Level-Signal zu liefern, was Spannungsunterschiede zwischen den LED-Pins erzeugt und die LED aufleuchten lässt.

```
digitalWrite(ledPin, HIGH);
```

Wenn das Level-Signal auf `LOW` geändert wird, wird das Signal von `ledPin` auf 0 V zurückgeführt, um die LED auszuschalten.

```
digitalWrite(ledPin, LOW);
```

Ein Intervall zwischen An- und Ausschalten ist erforderlich, damit Menschen die Veränderung sehen können, daher verwenden wir einen `delay(1000)`-Code, um den Controller 1000 ms lang nichts tun zu lassen.

```
delay(1000);
```

4.11 RGB-LED

4.11.1 Überblick

In dieser Lektion werden wir PWM nutzen, um eine RGB-LED verschiedene Farben anzeigen zu lassen. Wenn unterschiedliche PWM-Werte an den R-, G- und B-Pins der LED eingestellt werden, ändert sich deren Helligkeit. Wenn die drei verschiedenen Farben gemischt werden, können wir sehen, dass die RGB-LED unterschiedliche Farben aufleuchten lässt.

4.11.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

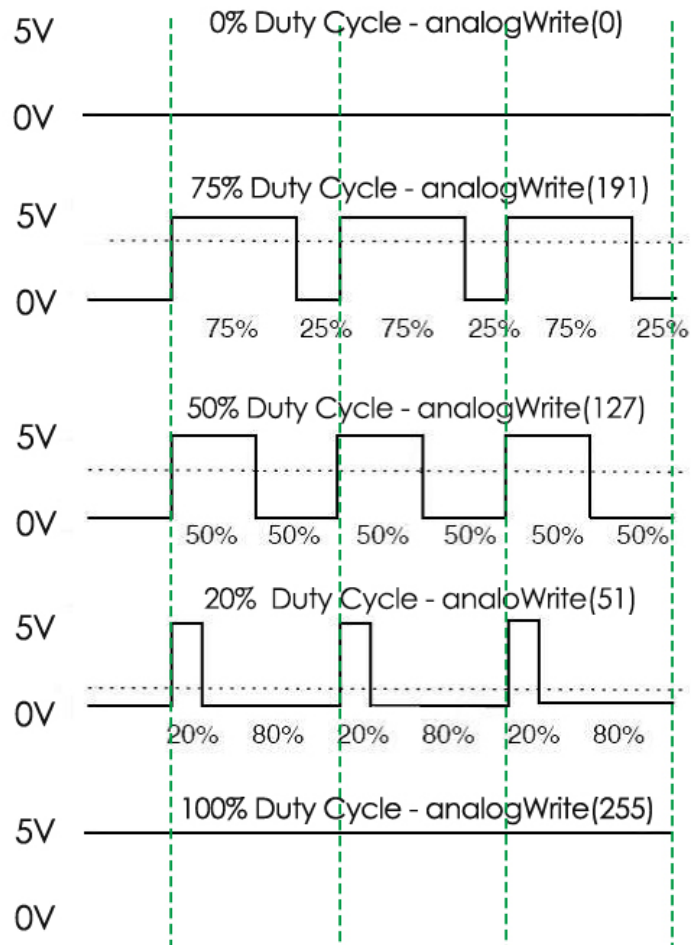
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>RGB-LED</i>	

4.11.3 PWM

Pulsweitenmodulation oder PWM ist eine Technik, um analoge Ergebnisse mit digitalen Mitteln zu erzielen. Digitale Steuerung wird verwendet, um eine Rechteckwelle zu erzeugen, ein Signal, das zwischen Ein und Aus geschaltet wird. Dieses Ein-Aus-Muster kann Spannungen zwischen voll Ein (5 Volt) und Aus (0 Volt) simulieren, indem der Anteil der Zeit, in der das Signal eingeschaltet ist, im Vergleich zur Zeit, in der das Signal ausgeschaltet ist, verändert wird. Die Dauer der „Eins-Zeit“ wird als Pulsbreite bezeichnet. Um unterschiedliche analoge Werte zu erhalten, ändern oder modulieren Sie diese Breite. Wenn Sie dieses Ein-Aus-Muster schnell genug mit einem Gerät wiederholen, z.B. einer LED, wäre es so: Das Signal ist eine konstante Spannung zwischen 0 und 5V, die die Helligkeit der LED steuert. (Siehe die PWM-Beschreibung auf der offiziellen Website von Arduino).

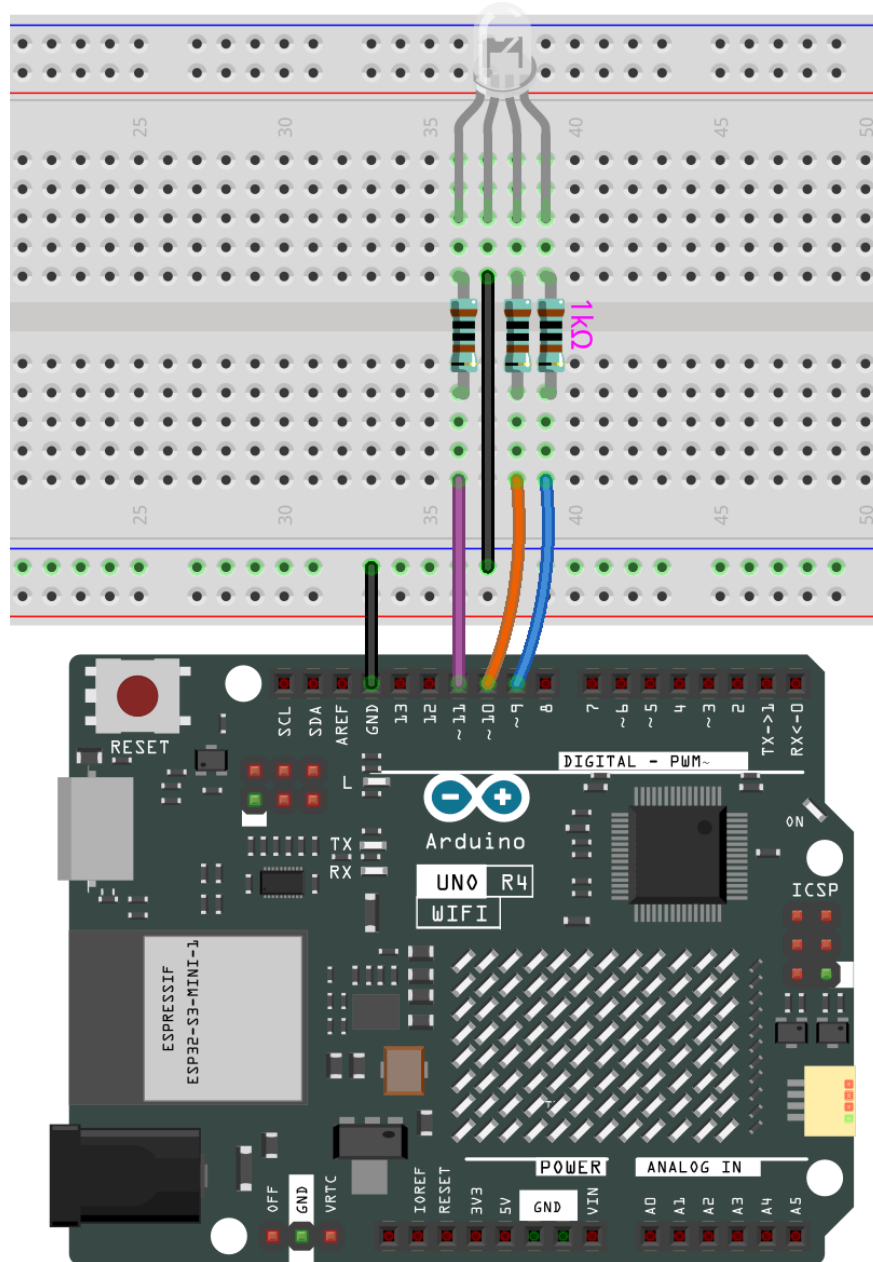
In der untenstehenden Grafik repräsentieren die grünen Linien einen regelmäßigen Zeitraum. Diese Dauer oder Periode ist das Inverse der PWM-Frequenz. Mit anderen Worten, bei einer PWM-Frequenz von etwa 500Hz würde jede grüne Linie 2 Millisekunden messen.



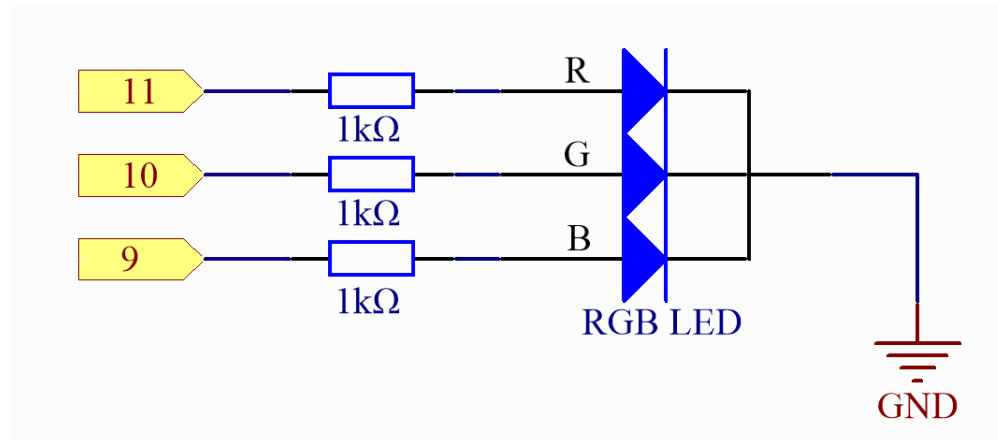
Ein Aufruf von `analogWrite()` erfolgt auf einer Skala von 0 - 255, so dass `analogWrite(255)` einen 100%igen Tastgrad (immer eingeschaltet) anfordert und `analogWrite(127)` einen 50%igen Tastgrad (die Hälfte der Zeit eingeschaltet) ist, zum Beispiel.

Sie werden feststellen, dass je kleiner der PWM-Wert ist, desto geringer wird der Wert sein, nachdem er in Spannung umgewandelt wurde. Dann wird die LED entsprechend dunkler. Daher können wir die Helligkeit der LED steuern, indem wir den PWM-Wert kontrollieren.

4.11.4 Verdrahtung



4.11.5 Schaltplan



4.11.6 Code

Bemerkung:

- Sie können die Datei `11-rgb_led.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\11-rgb_led` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem der Code erfolgreich hochgeladen wurde, werden Sie beobachten, dass die RGB-LED zunächst in einem kreisförmigen Muster in Rot, Grün und Blau blinkt. Dann wird sie in der Reihenfolge Rot, Orange, Gelb, Grün, Blau, Indigo und Lila blinken.

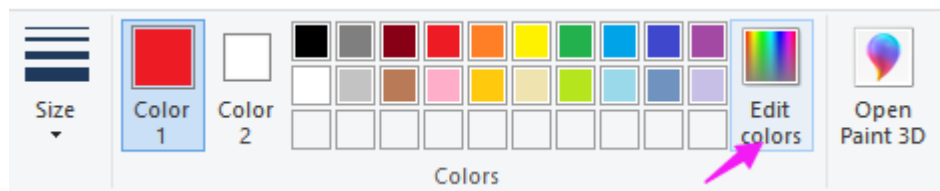
4.11.7 Code-Analyse

Farbe einstellen

Hier verwenden wir die Funktion `color()`, um die Farbe der RGB-LED einzustellen. Im Code wird sie so eingestellt, dass sie 7 verschiedene Farben blinkt.

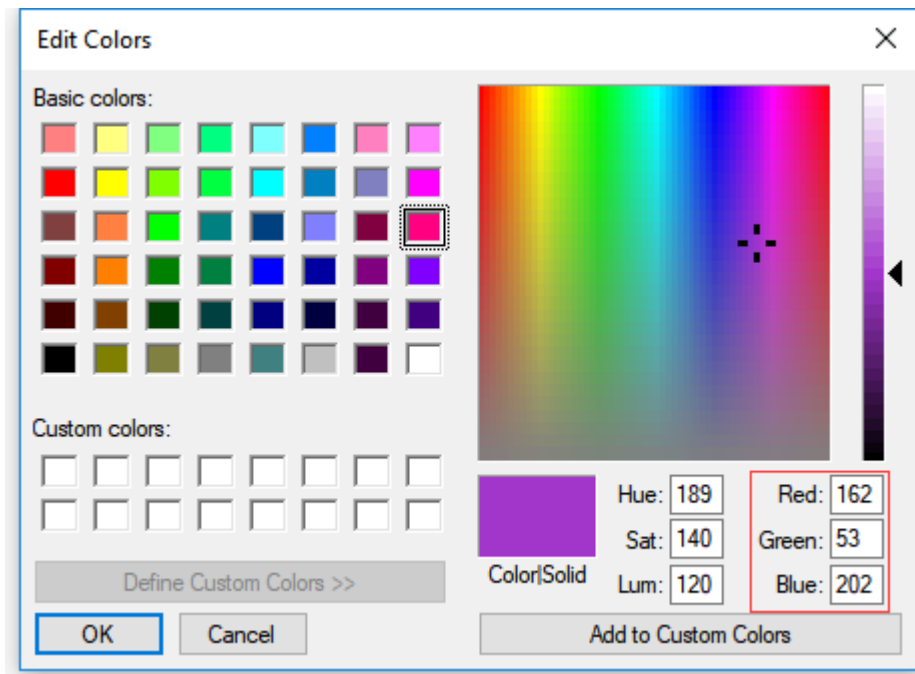
Sie können das Malprogramm auf Ihrem Computer verwenden, um den RGB-Wert zu erhalten.

1. Öffnen Sie das Malprogramm auf Ihrem Computer und klicken Sie auf Farben bearbeiten.



2. Wählen Sie eine Farbe aus, dann können Sie den RGB-Wert dieser Farbe sehen. Füllen Sie sie im Code aus.

Bemerkung: Aufgrund von Hardware- und Umweltfaktoren können die Farben auf Computerbildschirmen und RGB-LEDs variieren, auch wenn dieselben RGB-Werte verwendet werden.



```

void loop() // run over and over again
{
    // Basic colors:
    color(255, 0, 0); // turn the RGB LED red
    delay(1000); // delay for 1 second

    color(0,255, 0); // turn the RGB LED green
    delay(1000); // delay for 1 second

    color(0, 0, 255); // turn the RGB LED blue
    delay(1000); // delay for 1 second

    // Example blended colors:
    color(255,0,252); // turn the RGB LED red
    delay(1000); // delay for 1 second

    color(237,109,0); // turn the RGB LED orange
    delay(1000); // delay for 1 second

    color(255,215,0); // turn the RGB LED yellow
    .....
}

```

color() Funktion

```
void color (int red, int green, int blue)
// the color generating function

{

    analogWrite(redPin, red);

    analogWrite(greenPin, green);

    analogWrite(bluePin, blue);

}
```

Definieren Sie drei unsigned char Variablen, rot, grün und blau. Schreiben Sie deren Werte in redPin, greenPin und bluePin. Zum Beispiel erzeugt color(128,0,128) das Schreiben von 128 an redPin, 0 an greenPin und 128 an bluePin. Das Ergebnis ist dann das Blinken der LED in Lila.

analogWrite(): Schreibt einen analogen Wert (PWM-Welle) an einen Pin. Es hat nichts mit einem analogen Pin zu tun, sondern ist nur für PWM-Pins gedacht. Sie müssen nicht zuerst die Funktion pinMode() aufrufen, um den Pin als Ausgang zu setzen, bevor Sie analogWrite() aufrufen.

4.12 WS2812 RGB-LED-Streifen

4.12.1 Überblick

In dieser Lektion lernen Sie NeoPixel-LEDs und ihre Steuerung mit der FastLED-Bibliothek auf einem Arduino Uno R4 kennen. NeoPixel-LEDs werden in verschiedenen Anwendungen wie Heimdekoration, tragbaren Geräten und Eventbeleuchtung verwendet. Die FastLED-Bibliothek vereinfacht die Programmierung dieser LEDs. Hier wird eine Kette von 8 NeoPixel-LEDs an einen Arduino angeschlossen, und jede LED in der Reihe leuchtet kurz in blauer Farbe auf, bevor sie sich ausschaltet und zur nächsten LED in der Kette übergeht. Dieses einfache Beispiel kann als Grundlage für komplexere Lichtmuster oder interaktive Beleuchtungsprojekte dienen.

4.12.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

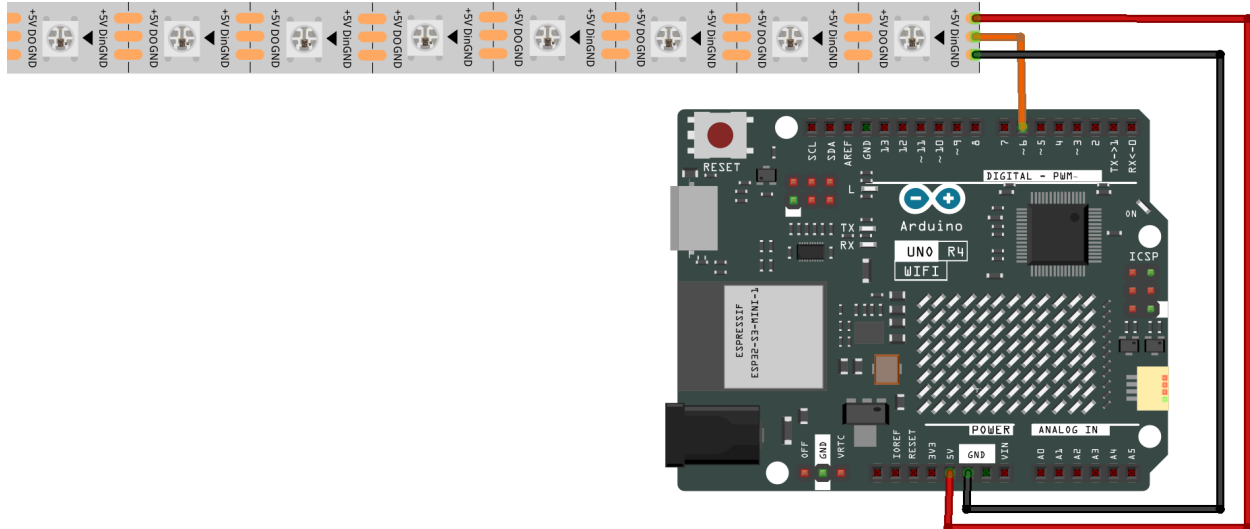
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

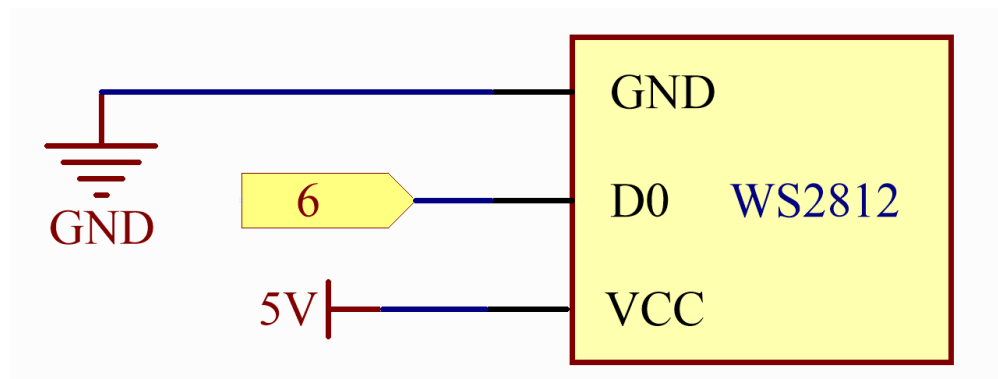
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>WS2812 RGB 8 LEDs Streifen</i>	

4.12.3 Verdrahtung



4.12.4 Schaltplan



4.12.5 Code

Bemerkung:

- Sie können die Datei 12-ws2812.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\12-ws2812 öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „FastLED“ und installieren Sie diese.

Nachdem der Code erfolgreich hochgeladen wurde, werden Sie sehen, wie jede LED in der Kette von 8 NeoPixel-LEDs nacheinander in blauer Farbe aufleuchtet. Das Programm wird diese Sequenz kontinuierlich durchlaufen, wobei jede

LED ausgeschaltet wird, bevor sie zur nächsten übergeht. Mit einer kurzen Verzögerung zwischen jeder LED erscheint der Beleuchtungseffekt wie ein entlang der Kette reisender blauer Punkt.

4.12.6 Code-Analyse

1. Bibliothek importieren und Konstanten einrichten

- Import der FastLED-Bibliothek, um ihre Funktionen zu nutzen.
- Definition der Anzahl der LEDs und des Datenpins, an den sie angeschlossen sind.

```
#include <FastLED.h> // Include FastLED library
#define NUM_LEDS 8    // Number of LEDs in the chain
#define DATA_PIN 6    // Data pin for LED control
```

2. LED-Array initialisieren

Erstellen eines Arrays vom Typ CRGB, um die Farbinformationen jeder LED zu speichern.

```
CRGB leds[NUM_LEDS]; // Array to hold LED color data
```

3. LEDs in Setup initialisieren

Verwendung von FastLED.addLeds, um die LEDs zu initialisieren.

```
void setup() {
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS); // Initialize LEDs
}
```

4. LEDs in Loop steuern

Durch jede LED iterieren, um sie blau zu setzen, anzuzeigen, zu löschen und dann zu verzögern.

- Das leds-Array dient als Farbpuffer für Ihren LED-Streifen. Jedes Element in diesem Array entspricht einer einzelnen LED auf Ihrem physischen Streifen, und sein Farbwert bestimmt die Farbe, die die LED anzeigen wird. Die Reihenfolge der Elemente im Array entspricht der Reihenfolge der LEDs auf dem Streifen, beginnend mit der ersten LED (die leds[0] entspricht) bis zur letzten LED. Um die Farbe einer bestimmten LED auf Ihrem Streifen zu ändern, ändern Sie einfach das entsprechende Element im leds-Array. Sie können verwenden oder Farben mit RGB einstellen (Grün als Beispiel, verwenden Sie leds[dot] = CRGB::Green oder leds[dot] = CRGB(0, 255, 0);).
- Die Funktion FastLED.show(); aktualisiert den LED-Streifen mit neuen Farbdaten und macht Änderungen sichtbar. Es ist wie das Drücken des „Veröffentlichen“-Knopfs für Ihren LED-Streifen, nachdem Sie Änderungen und Anpassungen im Code vorgenommen haben.

```
void loop() {
    for (int dot = 0; dot < NUM_LEDS; dot++) {
        leds[dot] = CRGB::Blue; // Set the current LED to blue
        FastLED.show();         // Update LEDs
        leds[dot] = CRGB::Black; // Clear the current LED
        delay(30);              // Wait for a short period before moving to the next
        ↪ LED
    }
}
```

4.13 7-Segment-Anzeige

4.13.1 Überblick

Eine 7-Segment-Anzeige ist ein Gerät, das Zahlen und Buchstaben darstellen kann. Sie besteht aus sieben in Parallelschaltung verbundenen LEDs. Verschiedene Buchstaben/Zahlen können angezeigt werden, indem die Pins der Anzeige mit der Stromquelle verbunden und die entsprechenden Pins aktiviert werden, wodurch die entsprechenden LED-Segmente eingeschaltet werden. In dieser Lektion lernen wir, wie man bestimmte Zeichen darauf anzeigt.

4.13.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

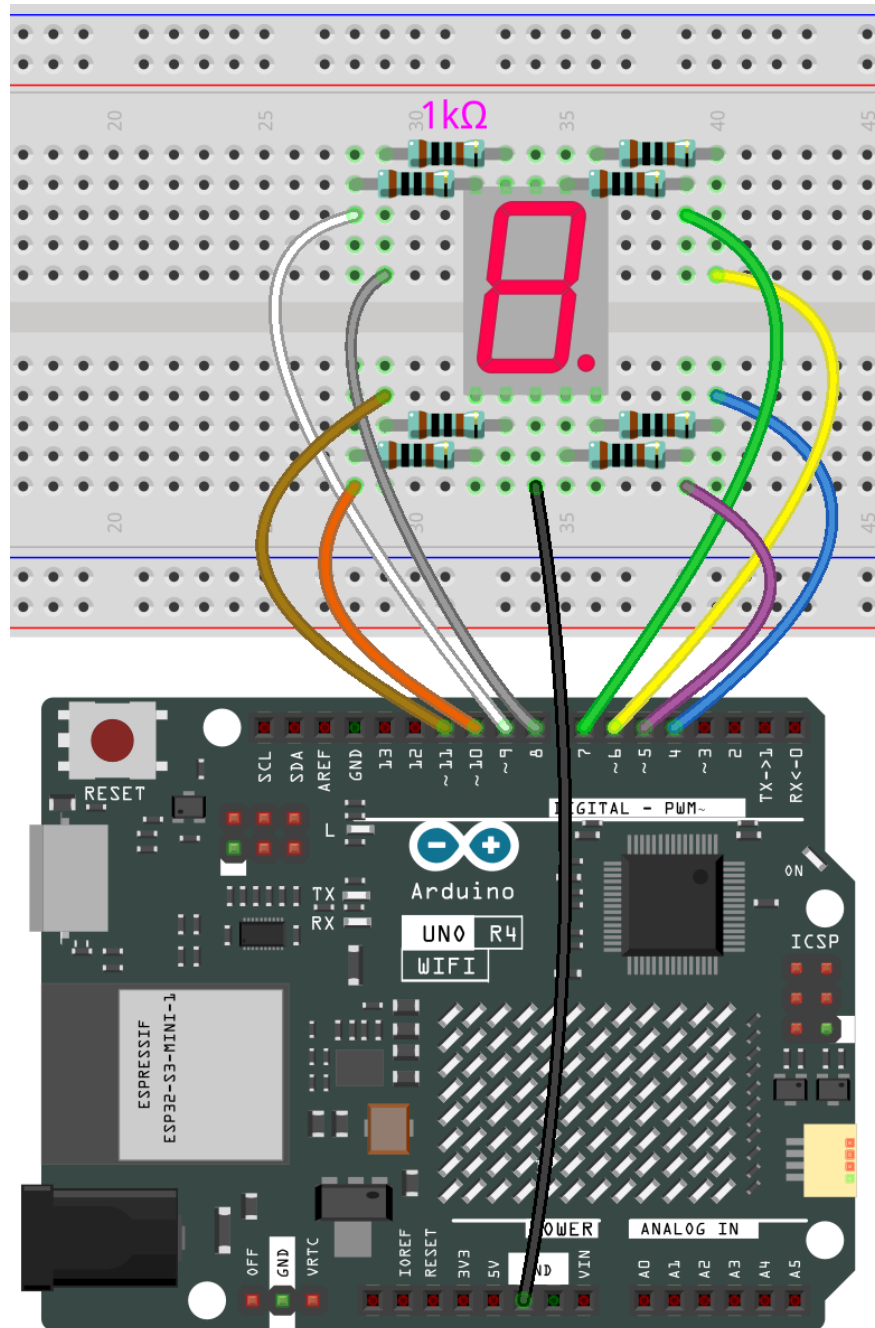
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

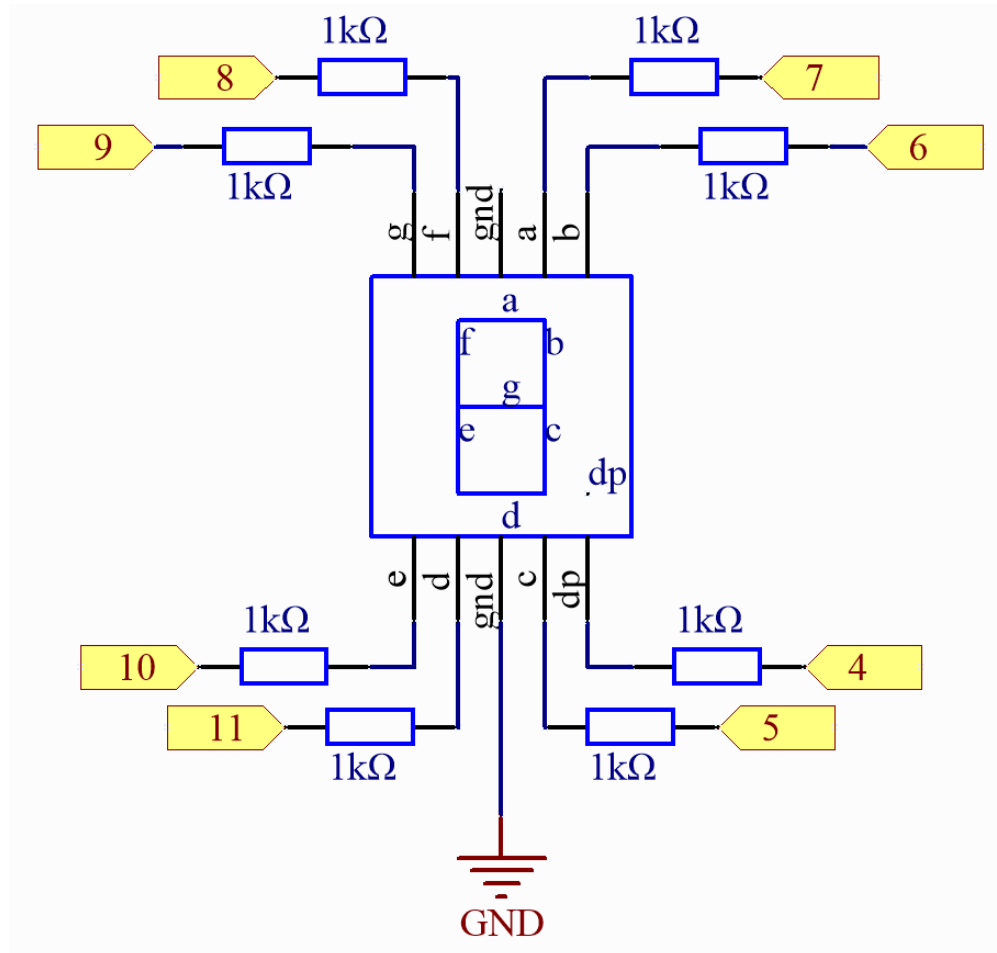
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>7-Segment-Anzeige</i>	

4.13.3 Verdrahtung



4.13.4 Schaltplan

In diesem Experiment verbinden Sie jeden Pin a-g der 7-Segment-Anzeige jeweils mit einem 1000-Ohm-Strombegrenzungswiderstand und dann mit Pin 4-11. GND verbindet sich mit GND. Durch die Programmierung können wir einen oder mehrere der Pins 4-11 als High-Level setzen, um die entsprechende(n) LED(s) zu beleuchten.



4.13.5 Code

Bemerkung:

- Sie können die Datei 13-7_segment.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\13-7_segment öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

4.13.6 Code-Analyse

Der Code könnte für dieses Experiment etwas lang erscheinen. Aber die Syntax ist einfach. Lassen Sie uns einen Blick darauf werfen.

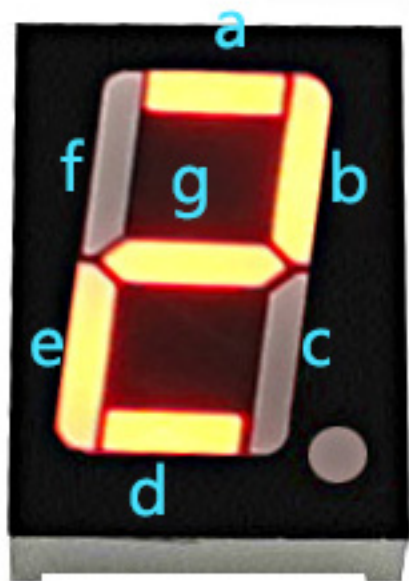
Aufrufen der Funktion in loop()

```
digital_1(); //diaplay 1 to the 7-segment  
  
delay(1000); //wait for a second  
  
digital_2(); //diaplay 2 to the 7-segment  
  
delay(1000); //wait for a second  
  
digital_3(); //diaplay 3 to the 7-segment  
  
delay(1000); //wait for a second  
  
digital_4(); //diaplay 4 to the 7-segment
```

Durch den Aufruf dieser Funktionen in der loop()-Funktion wird die 7-Segment-Anzeige 0-F anzeigen. Die Funktionen werden nachfolgend gezeigt. Nehmen wir digital_2() als Beispiel:

Detaillierte Analyse von digital_2()

```
void digital_2() //diaplay 2 to the 7-segment  
{  
    turnOffAllSegments();  
    digitalWrite(a, HIGH);  
    digitalWrite(b, HIGH);  
    digitalWrite(g, HIGH);  
    digitalWrite(e, HIGH);  
    digitalWrite(d, HIGH);  
}
```



Zuerst müssen wir verstehen, wie die Zahl **2** auf der 7-Segment-Anzeige erscheint. Dies wird erreicht, indem die Segmente a, b, d, e und g eingeschaltet werden. In der Programmierung werden Pins, die mit diesen Segmenten verbunden sind, auf ein hohes Niveau gesetzt, während c und f auf ein niedriges Niveau gesetzt werden. Wir beginnen mit der Funktion `turnOffAllSegments()`, um alle Segmente auszuschalten, und beleuchten dann die benötigten.

Nachdem dieser Teil ausgeführt wurde, zeigt die 7-Segment-Anzeige **2** an. Ähnlich ist es mit der Anzeige anderer Zeichen. Da die Buchstaben b und d in Großbuchstaben, nämlich **B** und **D**, gleich aussehen würden wie **8** und **0** auf der Anzeige, werden sie stattdessen in Kleinbuchstaben dargestellt.

4.14 I2C LCD1602

4.14.1 Überblick

In dieser Lektion erfahren Sie mehr über Flüssigkristallanzeigen (LCDs) mit einer I2C-Schnittstelle. Diese Art von LCDs wird in einer Vielzahl elektronischer Geräte verwendet, wie in Digitaluhren, Mikrowellenöfen, Armaturenbrettern in Autos und sogar in industriellen Geräten. Die I2C-Schnittstelle vereinfacht die Verkabelung und die Verbindungen, was sie sowohl für Hobbyisten als auch für Profis bequemer und effizienter macht.

4.14.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

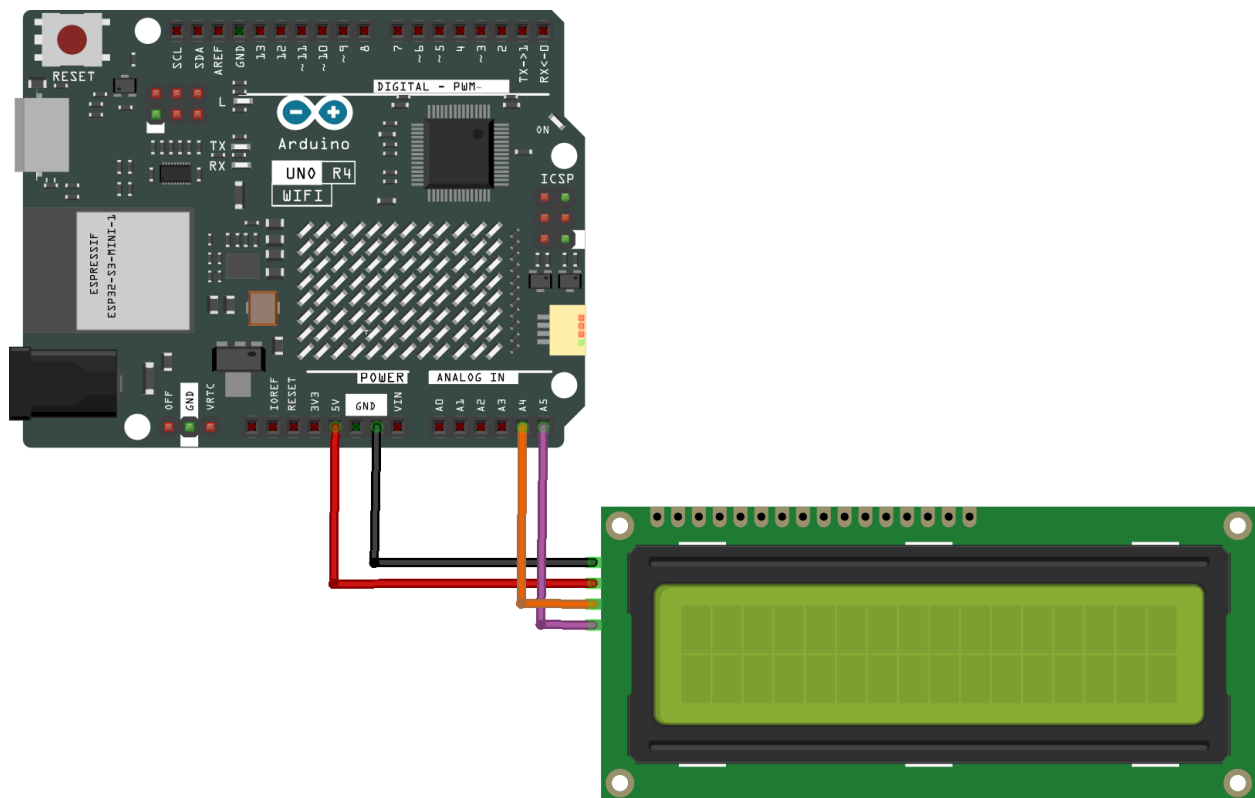
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

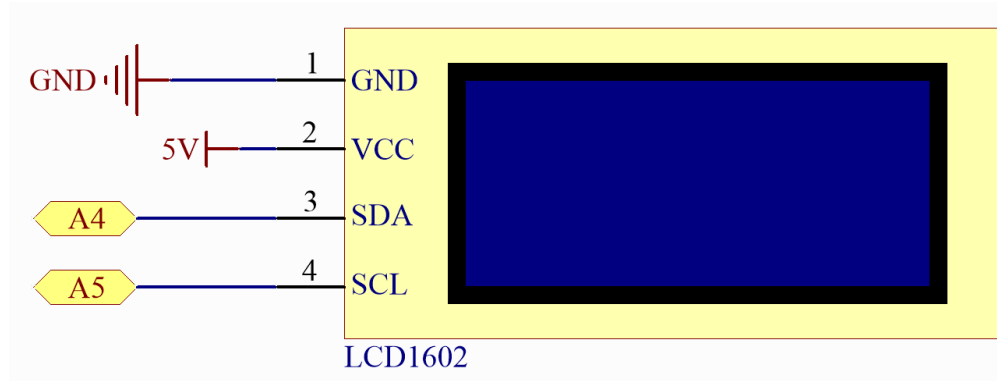
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>I2C LCD1602</i>	

4.14.3 Verdrahtung



4.14.4 Schaltplan



4.14.5 Code

Bemerkung:

- Sie können die Datei `14-i2c_lcd.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\14-i2c_lcd` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „**LiquidCrystal I2C**“ und installieren Sie diese.

Nachdem der Code erfolgreich auf den Arduino hochgeladen wurde, zeigt das Liquid Crystal Display (LCD) die Nachricht „Hello world!“ in der ersten Zeile und „LCD Tutorial“ in der zweiten Zeile an.

Bemerkung: Wenn das LCD nach dem Hochladen des Codes keine Zeichen anzeigt, können Sie den Kontrast einstellen, indem Sie das Potentiometer am I2C-Modul drehen, bis das LCD korrekt funktioniert.

4.14.6 Code-Analyse

1. Bibliothekseinbindung und LCD-Initialisierung: Die LiquidCrystal I2C-Bibliothek wird eingebunden, um Funktionen und Methoden für die LCD-Schnittstelle bereitzustellen. Danach wird ein LCD-Objekt mit der Klasse `LiquidCrystal_I2C` erstellt, wobei die I2C-Adresse, die Anzahl der Spalten und die Anzahl der Zeilen angegeben werden.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „**LiquidCrystal I2C**“ und installieren Sie diese.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

2. Setup-Funktion: Die `setup()`-Funktion wird einmal ausgeführt, wenn der Arduino startet. In dieser Funktion wird das LCD initialisiert, gelöscht und die Hintergrundbeleuchtung eingeschaltet. Anschließend werden zwei Nachrichten auf dem LCD angezeigt.

```
void setup() {  
  lcd.init();           // initialize the LCD  
  lcd.clear();          // clear the LCD display  
  lcd.backlight();      // Make sure backlight is on  
  
  // Print a message on both lines of the LCD.  
  lcd.setCursor(2, 0);  //Set cursor to character 2 on line 0  
  lcd.print("Hello world!");  
  
  lcd.setCursor(2, 1);  //Move cursor to character 2 on line 1  
  lcd.print("LCD Tutorial");  
}
```

4.15 OLED

4.15.1 Überblick

In dieser Lektion erfahren Sie mehr über OLED-Displays, die den SSD1306-Treiber verwenden. OLED-Displays (Organische Leuchtdioden) werden in verschiedenen elektronischen Geräten wie Smartwatches, Mobiltelefonen und sogar Fernsehern verwendet. Der SSD1306 ist ein Ein-Chip-CMOS-OLED/PLED-Treiber mit Controller für organische/polymerbasierte Leuchtdioden-Dot-Matrix-Grafikdisplaysysteme. Er bietet eine klare und deutliche visuelle Ausgabe durch organische, lichtemittierende Dioden, die bei Durchgang eines elektrischen Stroms leuchten.

Im bereitgestellten Code wird ein OLED-Display über das I2C-Protokoll mit einem Arduino-Board verbunden. Der Code verwendet die Adafruit SSD1306-Bibliothek zur Steuerung des Displays. Das Programm umfasst verschiedene Funktionalitäten wie:

1. Textanzeige: „Hello world!“ wird auf dem Bildschirm angezeigt.
2. Invertierter Text: Der Text „Hello world!“ wird in einem invertierten Farbschema dargestellt.
3. Schriftgröße: Der Text „Hello!“ wird mit einer vergrößerten Schriftgröße angezeigt.
4. Numerische Anzeige: Die Zahlen 123456789 werden angezeigt.
5. ASCII-Zeichen: Eine Reihe von ASCII-Zeichen wird angezeigt.
6. Scrollen: Text wird horizontal über das Display gescrollt.
7. Bitmap-Anzeige: Ein vordefiniertes Bitmap-Bild wird auf dem OLED-Bildschirm angezeigt.

Dieses OLED-Display kann in einer Vielzahl von Anwendungen eingesetzt werden, einschließlich digitaler Uhren, Mini-Spielkonsolen, Informationsanzeigen und so weiter. Es bietet eine großartige Möglichkeit, eine Benutzeroberfläche in kompakten und tragbaren Geräten bereitzustellen.

4.15.2 Benötigte Komponenten

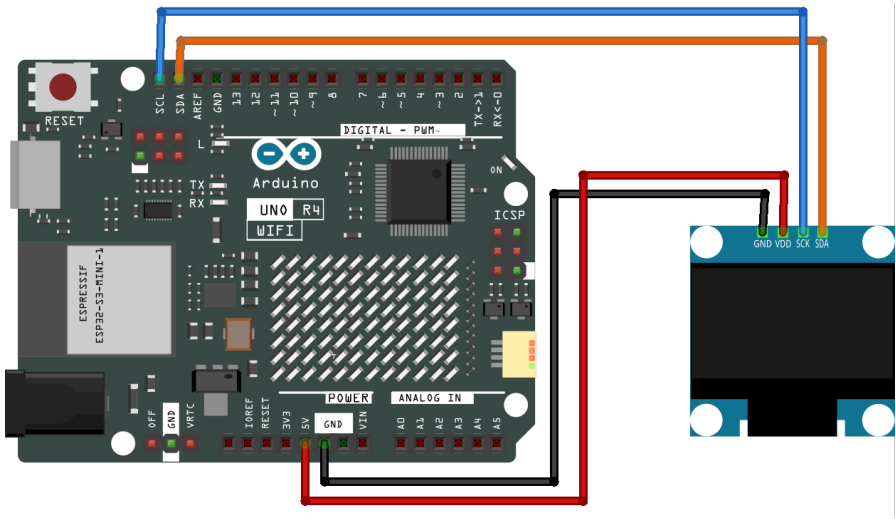
Für dieses Projekt benötigen wir die folgenden Komponenten.
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

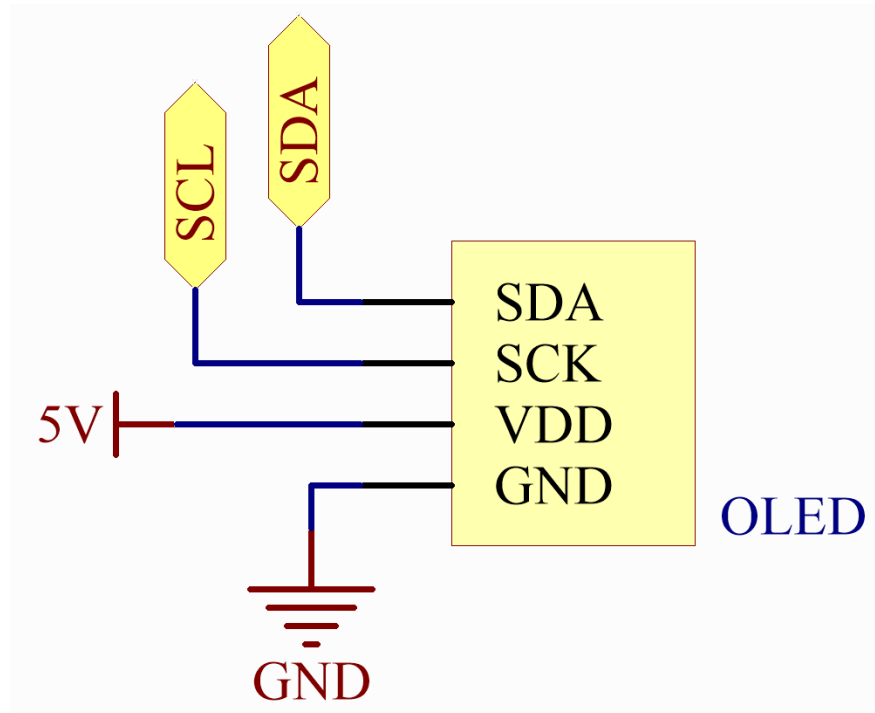
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Jumperkabel	
OLED-Display-Modul	

4.15.3 Verdrahtung



4.15.4 Schaltplan



4.15.5 Code

Bemerkung:

- Sie können die Datei `15-oled.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\15-oled` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „Adafruit SSD1306“ und „Adafruit GFX“ und installieren Sie diese.

4.15.6 Code-Analyse

1. **Einbindung der Bibliotheken und Anfangsdefinitionen:** Die notwendigen Bibliotheken für die Anbindung an das OLED werden eingebunden. Anschließend werden Definitionen bezüglich der Abmessungen und der I2C-Adresse des OLEDs bereitgestellt.
 - **Adafruit SSD1306:** Diese Bibliothek wurde entwickelt, um die Anbindung des SSD1306 OLED-Displays zu unterstützen. Sie bietet Methoden zur Initialisierung des Displays, Steuerung seiner Einstellungen und Anzeige von Inhalten.
 - **Adafruit GFX-Bibliothek:** Dies ist eine Kerngrafikbibliothek zum Anzeigen von Text, Erzeugen von Farben, Zeichnen von Formen usw. auf verschiedenen Bildschirmen, einschließlich OLEDs.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „**Adafruit SSD1306**“ und „**Adafruit GFX**“ und installieren Sie diese.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C
```

2. **Bitmap-Daten:** Bitmap-Daten zur Anzeige eines benutzerdefinierten Symbols auf dem OLED-Bildschirm. Diese Daten repräsentieren ein Bild in einem Format, das das OLED interpretieren kann.

Sie können dieses Online-Tool namens verwenden, um Ihr Bild in ein Array umzuwandeln.

Das Schlüsselwort `PROGMEM` zeigt an, dass das Array im Programmspeicher des Arduino-Mikrocontrollers gespeichert ist. Die Speicherung von Daten im Programmspeicher (`PROGMEM`) anstelle des RAMs kann hilfreich sein, wenn es sich um große Datenmengen handelt, die sonst zu viel Platz im RAM einnehmen würden.

```
static const unsigned char PROGMEM sunfounderIcon[] = {...};
```

3. **Setup-Funktion (Initialisierung und Anzeige):** Die `setup()`-Funktion initialisiert das OLED und zeigt eine Reihe von Mustern, Texten und Animationen an.

```
void setup() {
  ... // Serial initialization and OLED object initialization
  ... // Displaying various text, numbers, and animations
}
```

Ton

4.16 Aktiver Summer

4.16.1 Überblick

Der aktive Summer ist ein typisches digitales Ausgabegerät, das genauso einfach zu verwenden ist wie das Aufleuchten einer LED!

Das Kit enthält zwei Arten von Summern. Wir müssen den aktiven Summer verwenden. Drehen Sie sie um, die versiegelte Rückseite (nicht die offene Leiterplatte) ist die, die wir benötigen.



4.16.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

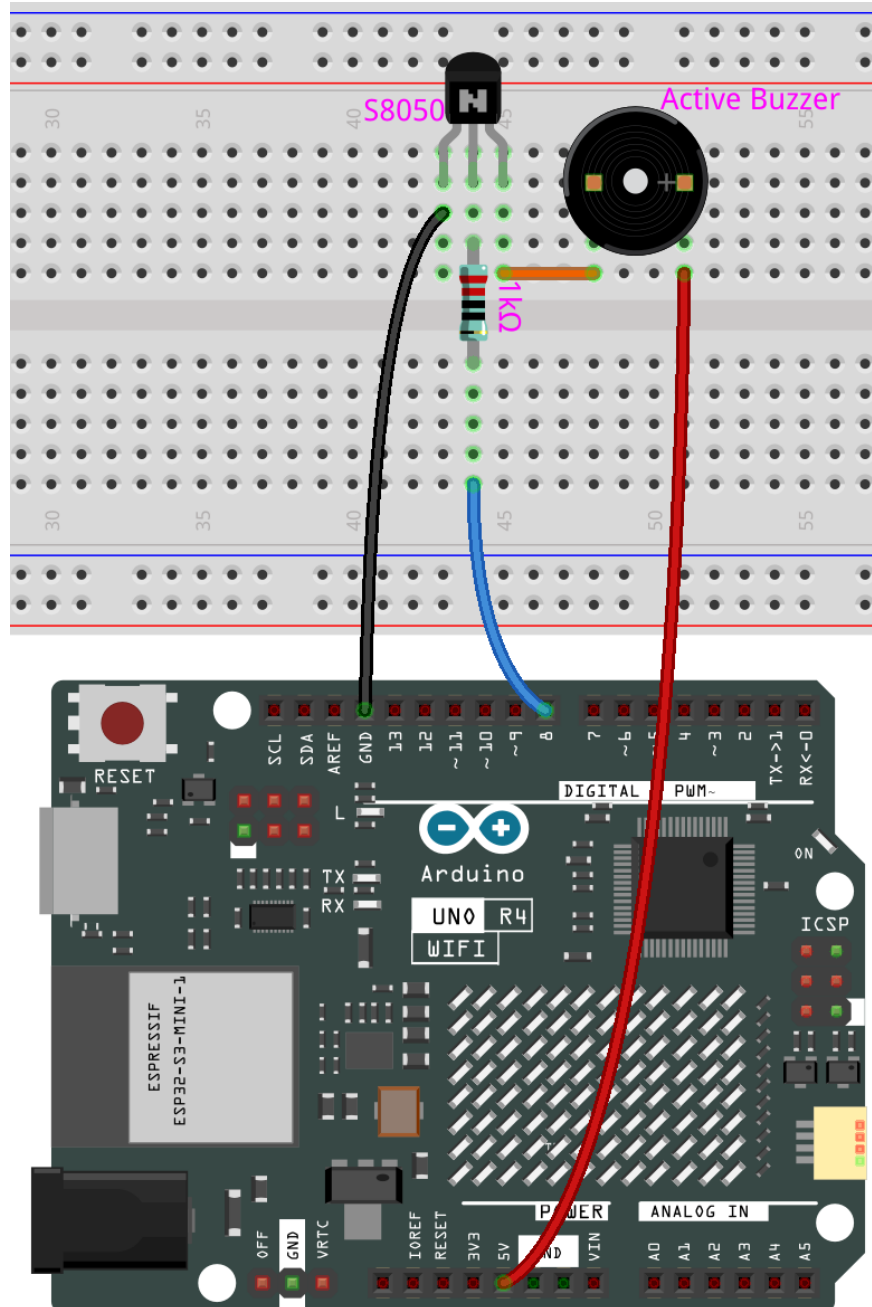
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

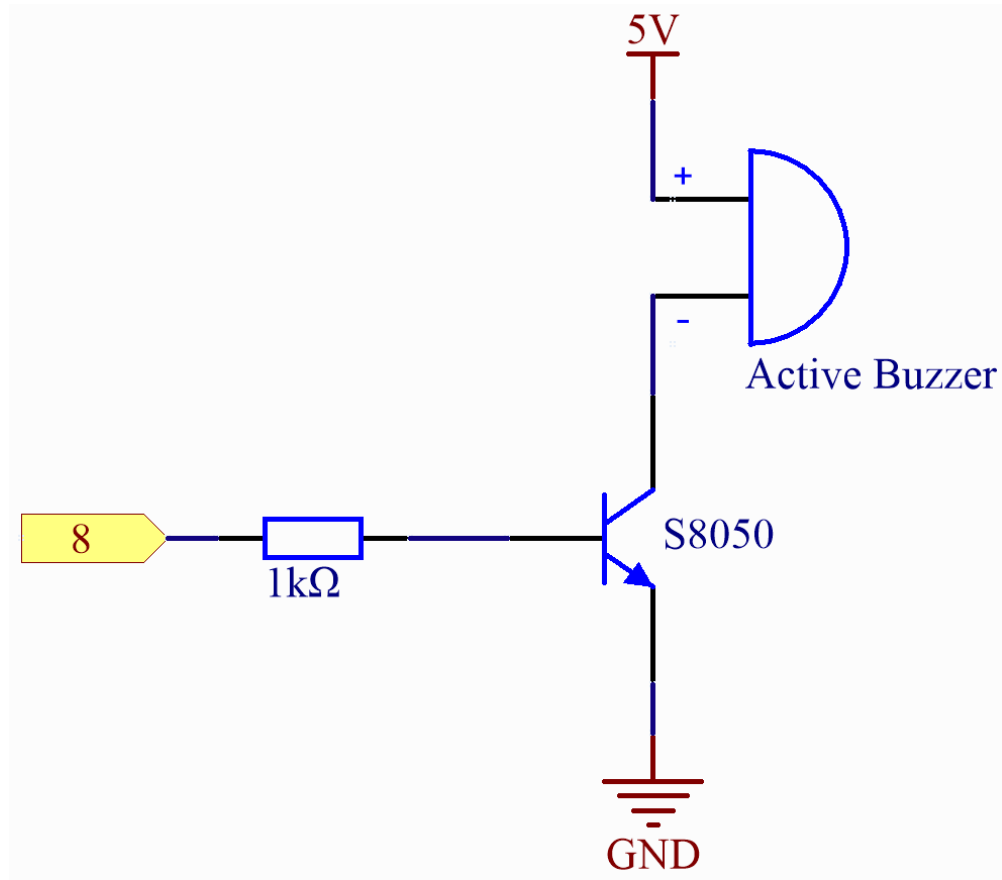
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Transistor</i>	
<i>Summer</i>	-

4.16.3 Verdrahtung

Bemerkung: Beim Anschließen des Summers achten Sie darauf, seine Pins zu überprüfen. Der längere Pin ist die Anode und der kürzere die Kathode. Es ist wichtig, sie nicht zu verwechseln, da sonst der Summer keinen Ton erzeugen wird.



4.16.4 Schaltplan



4.16.5 Code

Bemerkung:

- Sie können die Datei `16-active_buzzer.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\16-active_buzzer` öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Nachdem der Code erfolgreich hochgeladen wurde, hören Sie jede Sekunde einen Piepton.

4.17 Passiver Summer

4.17.1 Überblick

In diesem Projekt werden diese beiden Funktionen verwendet, um den passiven Summer zum Vibrieren zu bringen und Ton zu erzeugen. Die Funktion `tone()` erzeugt eine Rechteckwelle mit einer festgelegten Frequenz (und 50% Tastverhältnis) an einem Pin. Eine Dauer kann angegeben werden, oder die Welle setzt sich fort, bis `noTone()` aufgerufen wird. Ähnlich wie beim aktiven Summer nutzt auch der passive Summer elektromagnetische Induktion für seinen Betrieb. Der Unterschied besteht darin, dass ein passiver Summer keine eigene Schwingungsquelle hat und somit keinen Ton erzeugt, wenn Gleichstromsignale verwendet werden. Dies ermöglicht es jedoch, die eigene Schwingungsfrequenz des passiven Summers anzupassen und unterschiedliche Noten wie „do, re, mi, fa, sol, la, ti“ zu erzeugen.

4.17.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

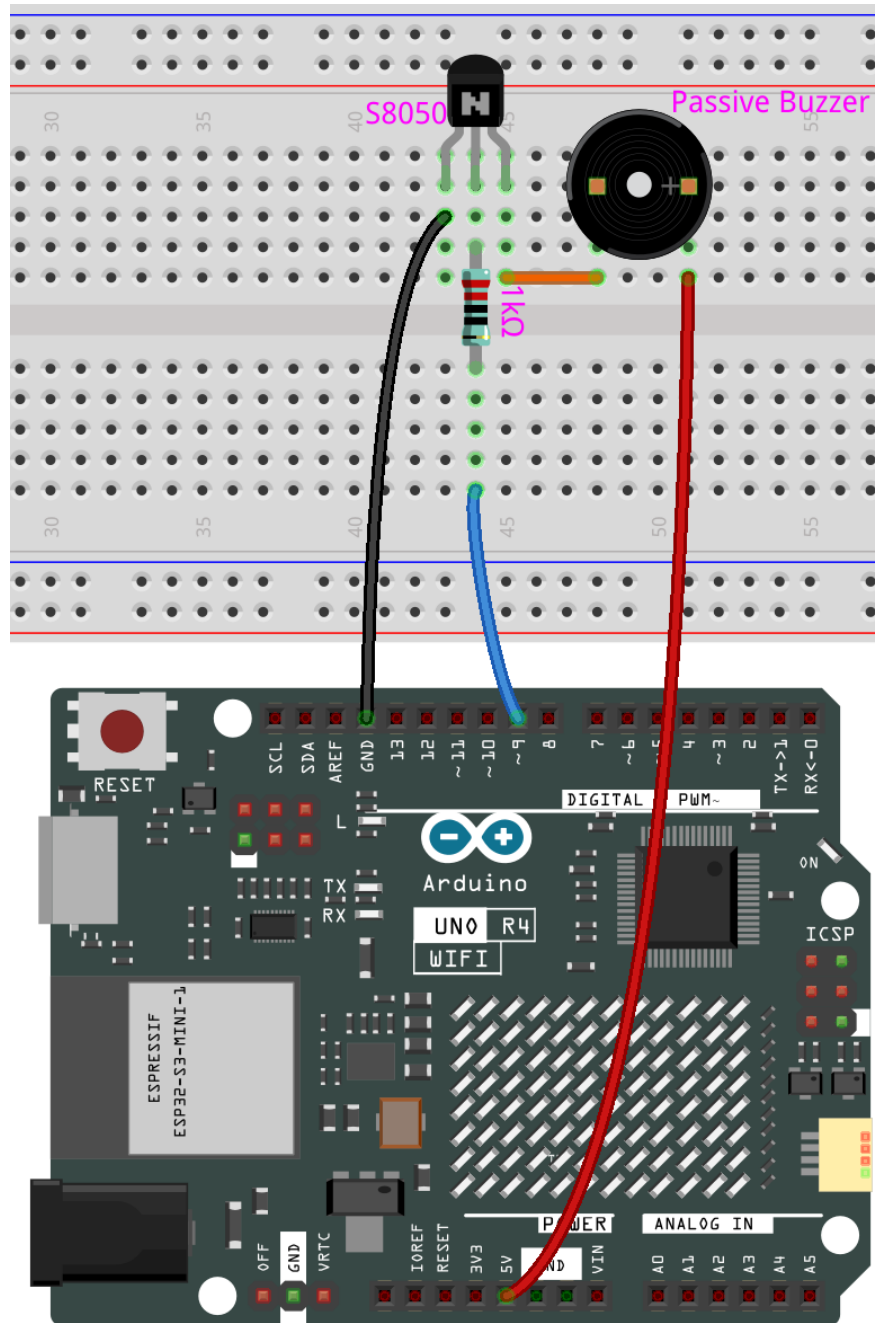
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

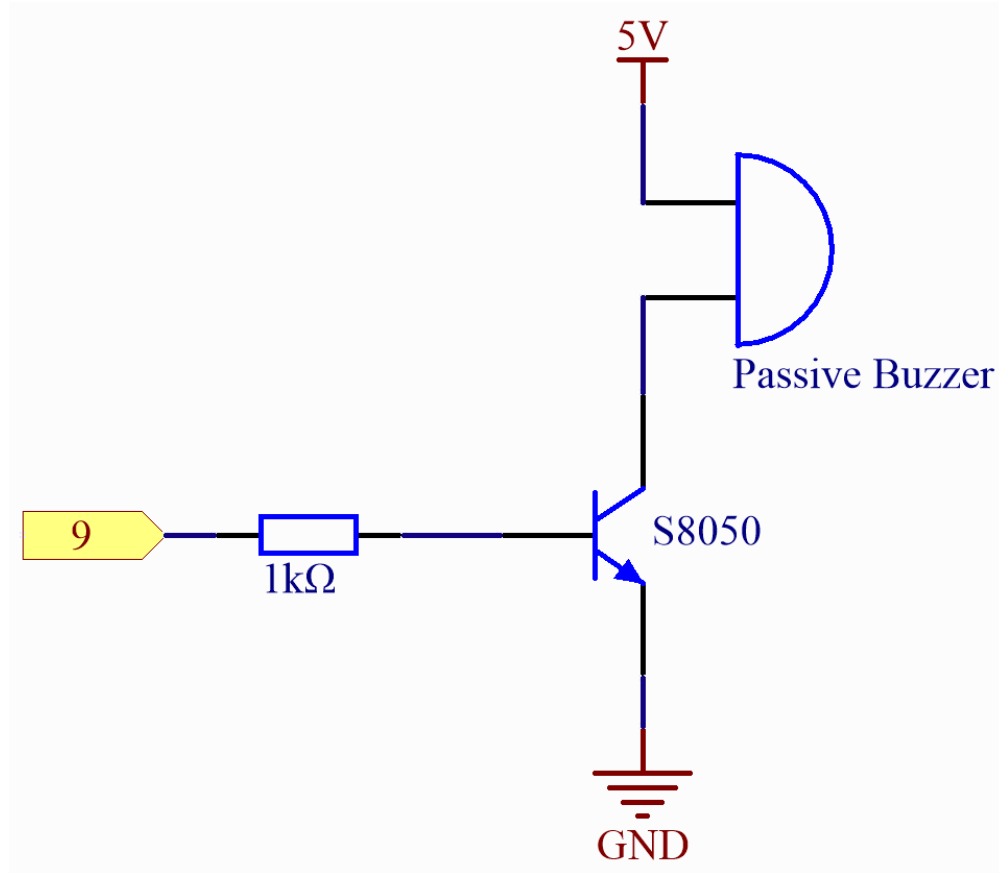
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Transistor</i>	
<i>Summer</i>	

4.17.3 Verdrahtung

Bemerkung: Beim Anschließen des Summers achten Sie darauf, seine Pins zu überprüfen. Der längere Pin ist die Anode und der kürzere die Kathode. Es ist wichtig, sie nicht zu verwechseln, da sonst der Summer keinen Ton erzeugen wird.



4.17.4 Schaltplan



4.17.5 Code

Bemerkung:

- Sie können die Datei `16-passive_buzzer.ino` direkt unter dem Pfad `elite-explorer-kit-main\basic_project\16-passive_buzzer` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem Sie die Codes auf das R4-Board hochgeladen haben, können Sie eine Melodie hören, die sieben Noten enthält.

4.17.6 Code-Analyse

1. Einbinden der Töne-Bibliothek: Diese Bibliothek stellt die Frequenzwerte für verschiedene musikalische Noten zur Verfügung, sodass Sie musikalische Notation in Ihrem Code verwenden können.

Bemerkung: Bitte platzieren Sie die Datei `pitch.h` im selben Verzeichnis wie den Code, um eine ordnungsgemäße Funktion zu gewährleisten.



```
#include "pitches.h"
```

2. Definition von Konstanten und Arrays:

- `buzzerPin` ist der digitale Pin am Arduino, an den der Summer angeschlossen ist.
- `melody[]` ist ein Array, das die Abfolge der zu spielenden Noten speichert.
- `noteDurations[]` ist ein Array, das die Dauer jeder Note in der Melodie speichert.

```
const int buzzerPin = 8;
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};
```

3. Abspielen der Melodie:

- Die `for`-Schleife iteriert über jede Note in der Melodie.
- Die Funktion `tone()` spielt eine Note auf dem Summer für eine bestimmte Dauer.
- Zwischen den Noten wird eine Verzögerung hinzugefügt, um sie zu unterscheiden.
- Die Funktion `noTone()` stoppt den Ton.

```
void setup() {
  for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(buzzerPin, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(buzzerPin);
  }
}
```

4. Leere `loop`-Funktion: Da die Melodie nur einmal im Setup gespielt wird, gibt es keinen Code in der `loop`-Funktion.
5. Fühlen Sie sich frei, mit dem Ändern der Noten und Dauern in den Arrays `melody[]` und `noteDurations[]` zu experimentieren, um Ihre eigenen Melodien zu kreieren. Wenn Sie interessiert sind, gibt es ein GitHub-Repository ([https://github.com/sunfounder/arduino-melody](#)), das Arduino-Codes für verschiedene Lieder bietet. Obwohl ihr Ansatz sich von diesem Projekt unterscheidet, können Sie deren Noten und Dauern als Referenz heranziehen.

4.18 Audio-Modul und Lautsprecher

4.18.1 Überblick

In dieser Lektion lernen Sie das Audio-Modul und den Lautsprecher in Verbindung mit einem Arduino Uno-Board kennen. Diese Komponenten werden in verschiedenen elektronischen Anwendungen eingesetzt, darunter in musikalischen Spielzeugen, DIY-Soundsystemen, Alarmen und sogar in anspruchsvollen Musikinstrumenten. Durch die Kombination eines Arduino mit einem Audio-Modul und einem Lautsprecher können Sie einen einfachen, aber effektiven Melodiespieler erstellen.

4.18.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

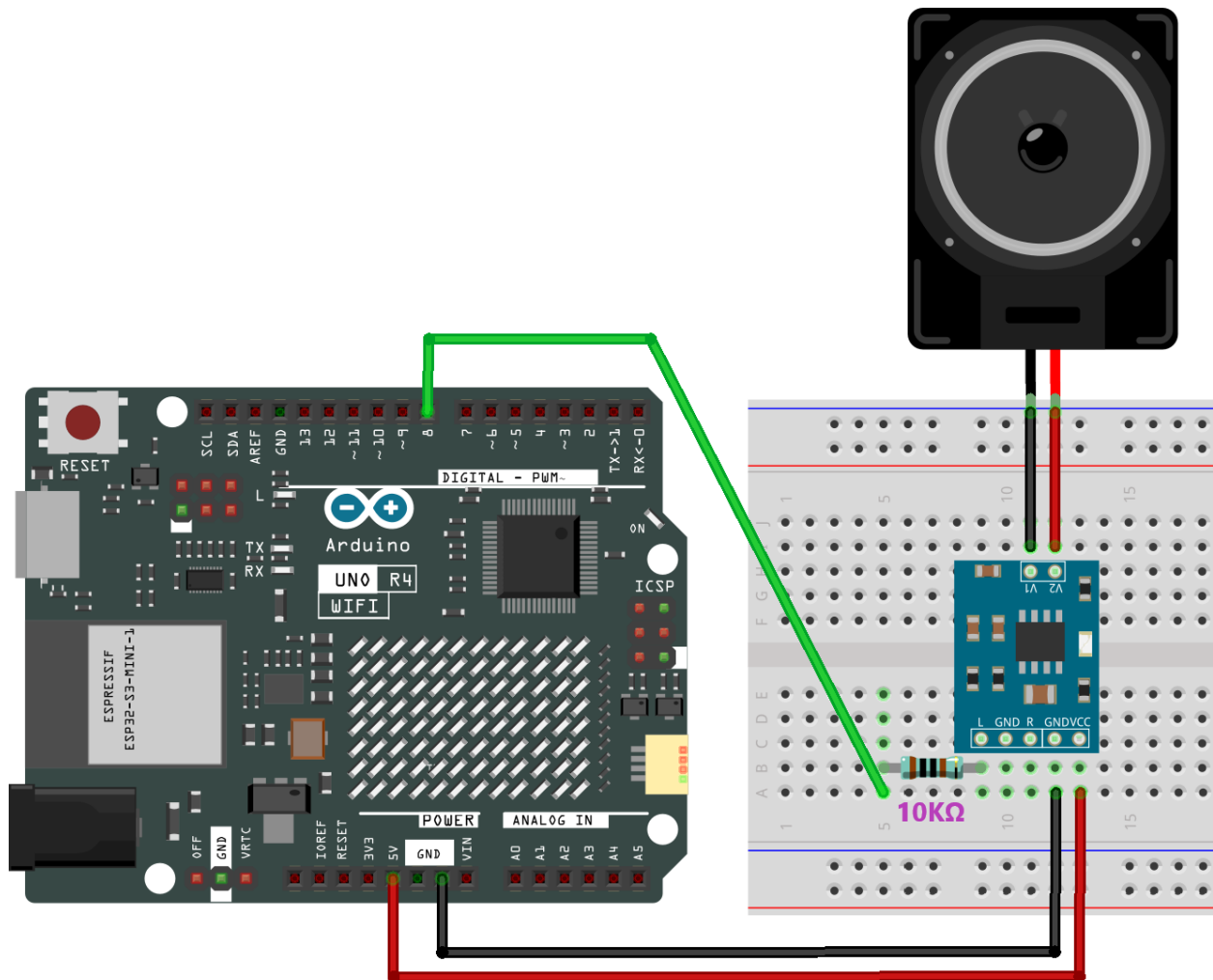
Sie können die Komponenten auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Audio-Modul und Lautsprecher</i>	-

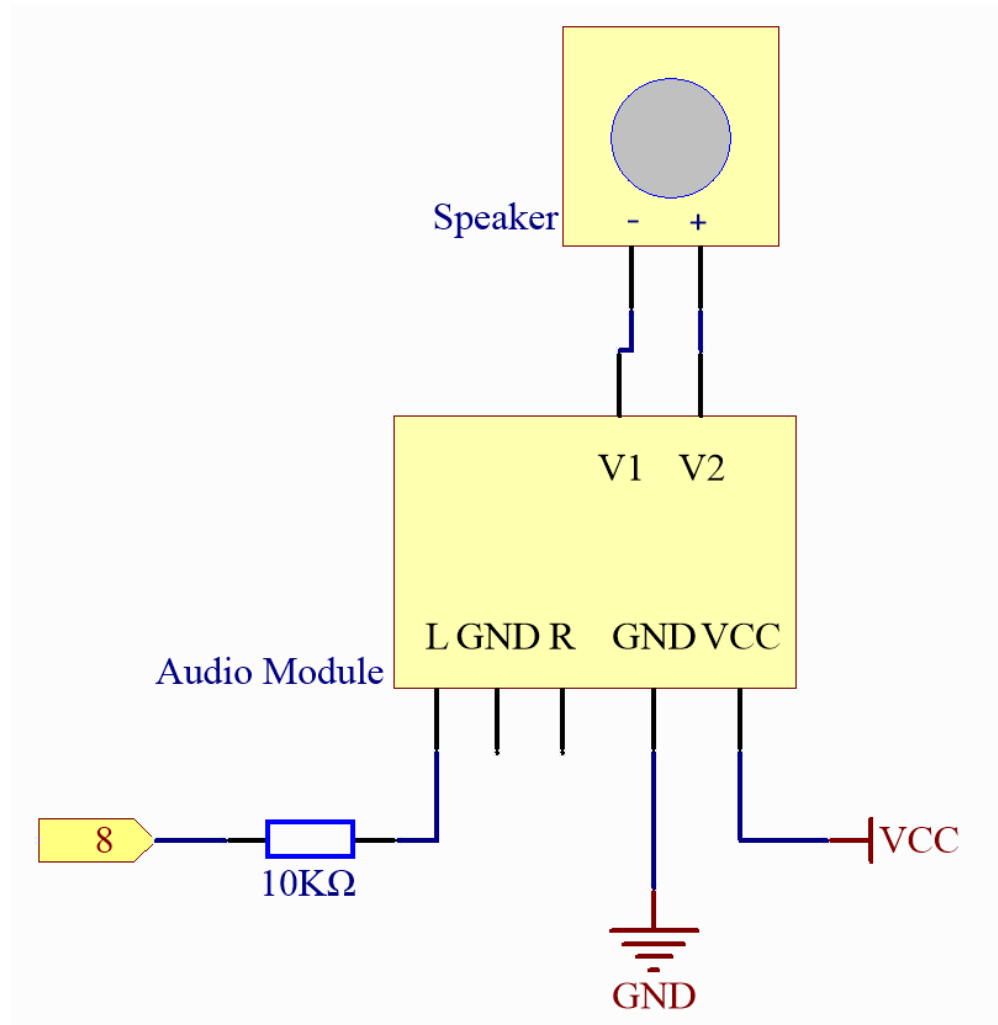
4.18.3 Verdrahtung

Da es sich um einen Mono-Verstärker handelt, können Sie Pin 8 an den L- oder R-Pin des Audioverstärkermoduls anschließen.

Der 10K-Widerstand wird verwendet, um hochfrequentes Rauschen zu reduzieren und die Lautstärke zu verringern. Er bildet zusammen mit der parasitären Kapazität des DAC und des Audioverstärkers einen RC-Tiefpassfilter. Dieser Filter verringert die Amplitude von hochfrequenten Signalen und reduziert so effektiv hochfrequentes Rauschen. Daher macht das Hinzufügen des 10K-Widerstands die Musik leiser und eliminiert unerwünschtes hochfrequentes Rauschen.



4.18.4 Schaltplan



4.18.5 Code

Bemerkung:

- Sie können die Datei 17-speaker.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\17-speaker öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Sobald Sie die Codes auf das R4-Board hochgeladen haben, können Sie eine Melodie hören, die sieben Noten enthält.

4.18.6 Code-Analyse

1. Einbindung der Tonhöhen-Bibliothek: Diese Bibliothek liefert die Frequenzwerte für verschiedene musikalische Noten und ermöglicht es Ihnen, musikalische Notation in Ihrem Code zu verwenden.

Bemerkung: Bitte platzieren Sie die Datei `pitches.h` im selben Verzeichnis wie den Code, um eine ordnungsgemäße Funktion zu gewährleisten.



```
#include "pitches.h"
```

2. Definition von Konstanten und Arrays:

- `speakerPin` ist der digitale Pin am Arduino, an den der Lautsprecher angeschlossen ist.
- `melody[]` ist ein Array, das die Reihenfolge der zu spielenden Noten speichert.
- `noteDurations[]` ist ein Array, das die Dauer jeder Note in der Melodie speichert.

```
const int speakerPin = 8;
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};
```

3. Abspielen der Melodie:

- Die `for`-Schleife iteriert über jede Note in der Melodie.
- Die Funktion `tone()` spielt eine Note auf dem Lautsprecher für eine bestimmte Dauer.
- Zwischen den Noten wird eine Pause hinzugefügt, um sie zu unterscheiden.
- Die Funktion `noTone()` stoppt den Ton.

```
void setup() {
  for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(speakerPin, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(speakerPin);
  }
}
```

4. Leere Loop-Funktion: Da die Melodie nur einmal im Setup gespielt wird, gibt es keinen Code in der Loop-Funktion.

5. Experimentieren Sie frei mit der Änderung der Noten und Dauern in den Arrays `melody[]` und `noteDurations[]`, um Ihre eigenen Melodien zu kreieren. Wenn Sie interessiert sind, gibt es ein GitHub-Repository ([https://github.com/SunFounder-Arduino-Enthusiast/101_LED_Basics](#)), das Arduino-Codes für das Spielen verschiedener Lieder bietet. Obwohl ihr Ansatz sich von diesem Projekt unterscheidet, können Sie deren Noten und Dauern als Referenz heranziehen.

Steuerung

4.19 Taster

4.19.1 Überblick

In dieser Lektion lernen Sie, wie Sie eine LED mit einem Arduino und einem Taster steuern können. Taster und LEDs sind grundlegende Komponenten in einer Vielzahl elektronischer Geräte, wie Fernbedienungen, Taschenlampen und interaktiven Installationen. In dieser Einrichtung wird ein Taster als Eingabegerät verwendet, um den Zustand einer LED zu steuern, die als Ausgabegerät dient.

Der Taster ist mit Pin 12 auf dem Arduino Uno R4 Board verbunden, und die LED ist mit Pin 13 verbunden. Wenn der Taster gedrückt wird, sendet er ein Signal an den Arduino, das die LED einschaltet. Umgekehrt, wenn der Taster losgelassen wird, erlischt die LED. Dieser einfache, aber effektive Mechanismus kann die Grundlage für komplexere Projekte sein, wie zum Beispiel Hausautomationssysteme, interaktive Displays und vieles mehr.

Am Ende dieser Lektion werden Sie verstehen, wie man Eingaben von einem Taster liest und sie verwendet, um eine LED zu steuern, und damit ein grundlegendes Verständnis für Ein-/Ausgabeoperationen mit Arduino erlangen.

4.19.2 Benötigte Komponenten

In diesem Projekt benötigen wir folgende Komponenten.

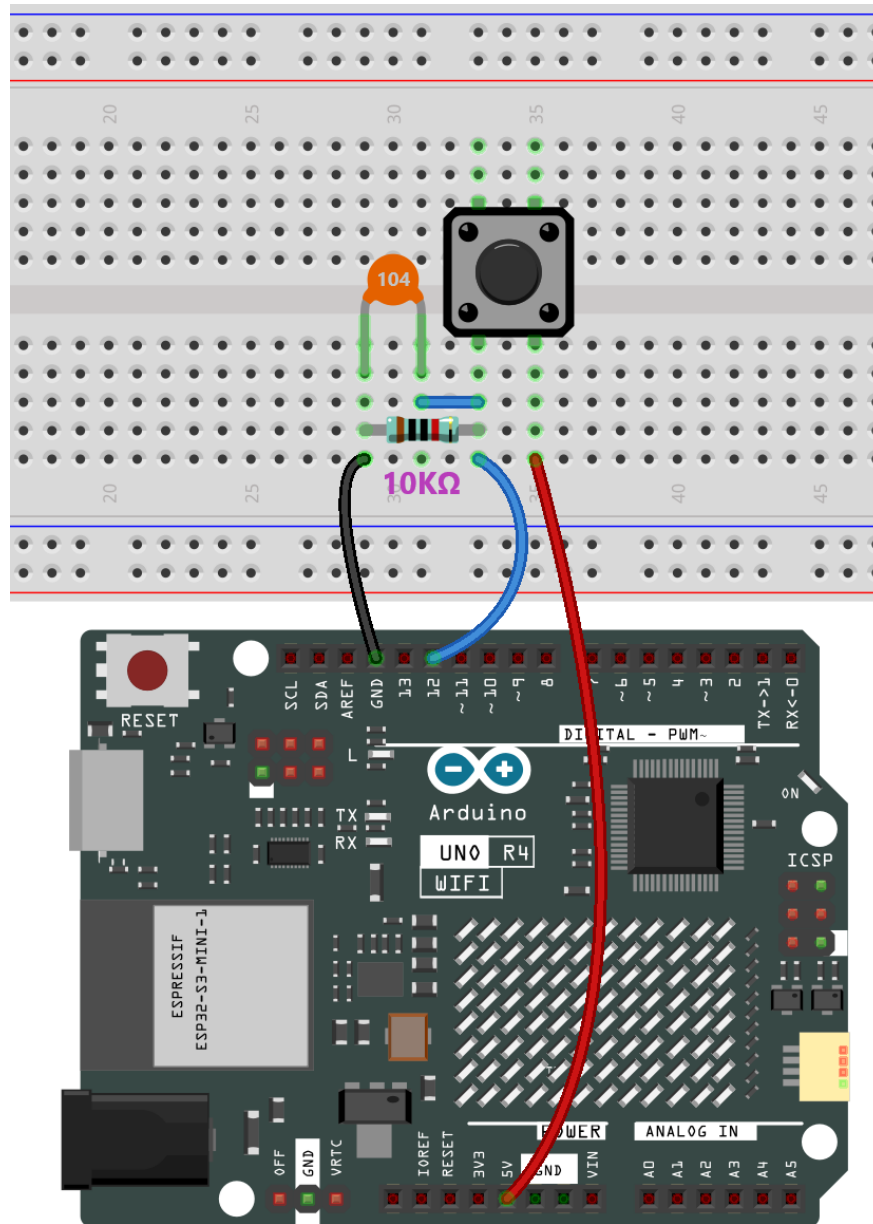
Es ist definitiv bequem, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können diese auch einzeln über die untenstehenden Links kaufen.

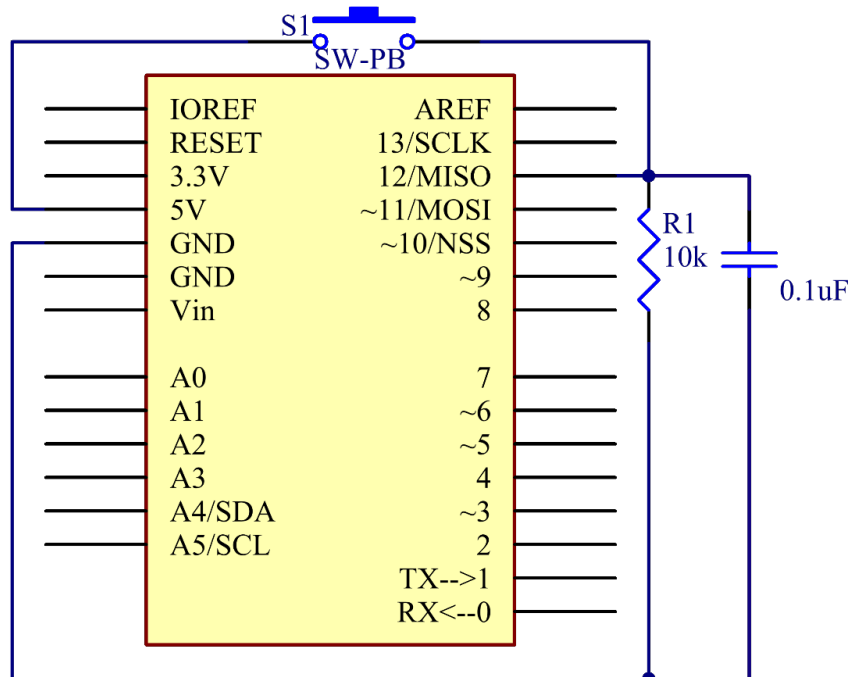
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Kondensator</i>	
<i>Knopf</i>	

4.19.3 Verdrahtung



4.19.4 Schaltplan

Verbinden Sie ein Ende des Tasters mit Pin 12, der mit einem Pull-Down-Widerstand und einem 0,1µF (104) Kondensator verbunden ist (um Zittern zu eliminieren und ein stabiles Level auszugeben, wenn der Taster arbeitet). Verbinden Sie das andere Ende des Widerstands mit GND und einen der Pins am anderen Ende des Tasters mit 5V. Wenn der Taster gedrückt wird, ist Pin 12 auf 5V (HIGH) und setzt gleichzeitig Pin 13 (integriert mit einer LED) auf High. Dann lassen Sie den Taster los (Pin 12 wechselt auf LOW) und Pin 13 ist Low. So werden wir sehen, dass die LED aufleuchtet und erlischt, wenn der Taster gedrückt und losgelassen wird.



4.19.5 Code

Bemerkung:

- Sie können die Datei 18-button.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\18-button öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

4.19.6 Code-Analyse

1. Konstanten und Variablen definieren

In diesem Abschnitt werden die Pinnummern für den Taster und die LED definiert. Außerdem wird eine Variable `buttonState` deklariert, um den aktuellen Zustand des Tasters zu speichern.

```
const int buttonPin = 12;
const int ledPin = 13;
int buttonState = 0;
```

2. Setup-Funktion

Die `setup()`-Funktion wird einmal ausgeführt, wenn das Arduino-Board startet. Die Pin-Modi für den Taster und die LED werden mit der Funktion `pinMode` eingestellt.

```
void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
```

3. Hauptschleife

Die `loop()`-Funktion wird wiederholt ausgeführt. Innerhalb dieser Schleife wird die Funktion `digitalRead()` verwendet, um den Zustand des Tasters zu lesen. Je nachdem, ob der Taster gedrückt ist oder nicht, wird die LED ein- oder ausgeschaltet.

```
void loop() {  
  buttonState = digitalRead(buttonPin);  
  if (buttonState == HIGH) {  
    digitalWrite(ledPin, HIGH);  
  } else {  
    digitalWrite(ledPin, LOW);  
  }  
}
```

4.20 Potentiometer

4.20.1 Überblick

In dieser Lektion sehen wir uns an, wie wir die Leuchtkraft einer LED mit einem Potentiometer verändern und die Daten des Potentiometers im Seriellen Monitor ablesen, um seine Wertänderungen zu beobachten.

4.20.2 Benötigte Komponenten

In diesem Projekt benötigen wir die folgenden Komponenten.

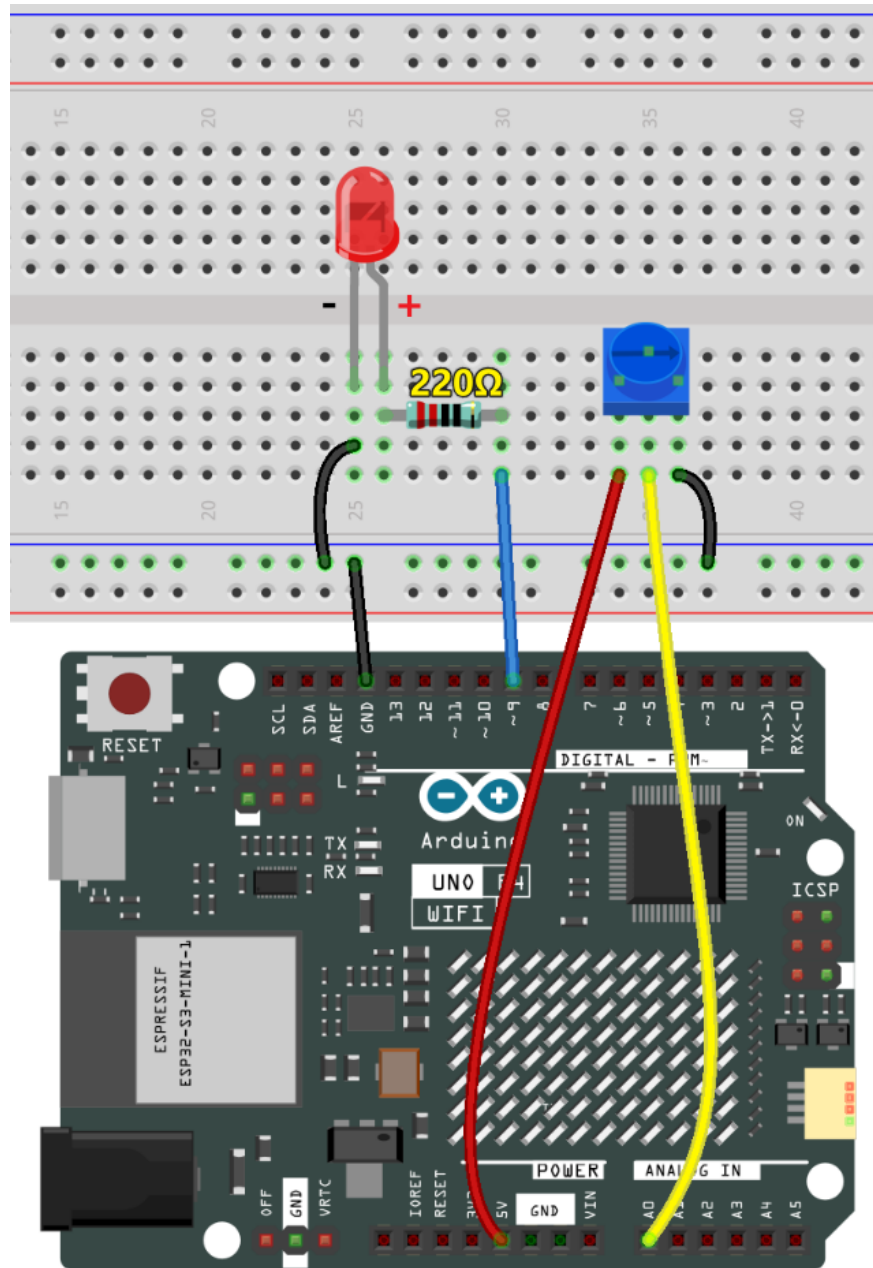
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

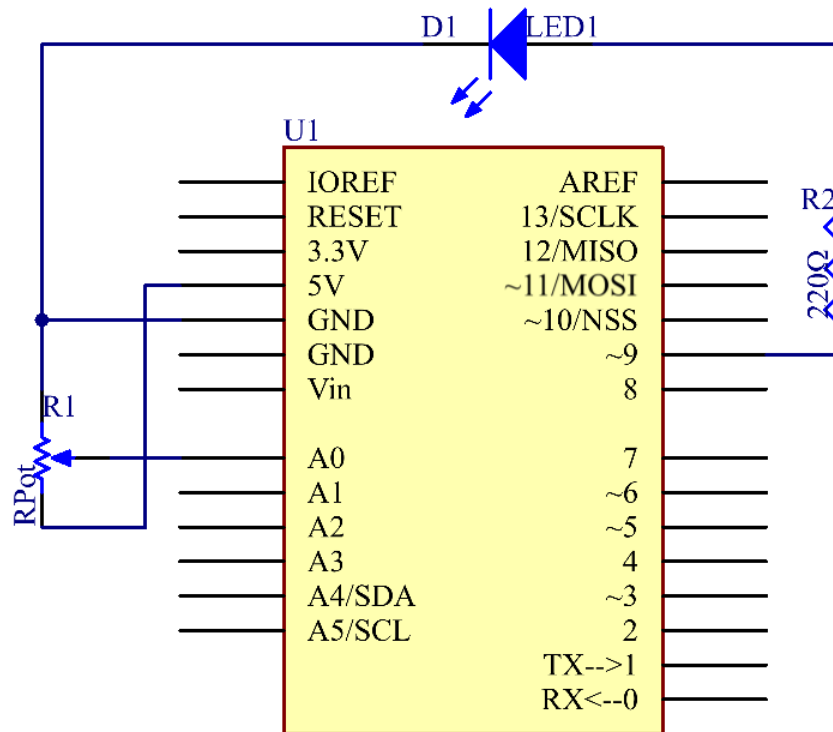
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Potentiometer</i>	

4.20.3 Verdrahtung



4.20.4 Schaltplan

In diesem Experiment wird das Potentiometer als Spannungsteiler verwendet, was bedeutet, dass Geräte an alle drei Pins angeschlossen werden. Verbinden Sie den mittleren Pin des Potentiometers mit Pin A0 und die anderen beiden Pins mit 5V bzw. GND. Daher liegt die Spannung des Potentiometers zwischen 0-5V. Drehen Sie den Knopf des Potentiometers, und die Spannung an Pin A0 wird sich ändern. Dann wird diese Spannung mit dem AD-Wandler im Steuerboard in einen digitalen Wert (0-1024) umgewandelt. Durch Programmierung können wir den umgewandelten digitalen Wert nutzen, um die Helligkeit der LED auf dem Steuerboard zu steuern.



4.20.5 Code

Bemerkung:

- Sie können die Datei 19-potentiometer.ino direkt unter dem Pfad elite-explorer-kit-main\basic_project\19-potentiometer öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem der Code auf das Uno-Board hochgeladen wurde, kann der serielle Monitor geöffnet werden, um die gelesenen Werte des Potentiometers zu beobachten. Wenn Sie am Knopf des Potentiometers drehen, ändert sich der gelesene Wert entsprechend. Der rohe Analogwert des Potentiometers reicht von (0) bis (1023). Gleichzeitig skaliert der Code diesen Wert in einen Bereich von (0) bis (255), der auch auf dem seriellen Monitor angezeigt wird. Dieser skalierte Wert wird dann verwendet, um die Helligkeit der angeschlossenen LED zu steuern. Die LED wird heller oder dunkler, je nach skaliertem Wert. Es ist anzumerken, dass der theoretische Bereich des Potentiometers (0) bis (1023) beträgt, der tatsächliche Bereich kann jedoch aufgrund von Hardwaretoleranzen leicht variieren.

4.20.6 Code-Analyse

1. Initialisierung und Einrichtung (Einstellen der Pin-Modi und Initialisieren der seriellen Kommunikation)

Bevor wir in die Schleife einsteigen, definieren wir, welche Pins wir verwenden und initialisieren die serielle Kommunikation.

```
const int analogPin = 0; // Analog input pin connected to the potentiometer
const int ledPin = 9;    // Digital output pin connected to the LED

void setup() {
  Serial.begin(9600); // Initialize serial communication with a baud rate of 9600
}
```

2. Lesen des Analogeingangs (Daten vom Potentiometer abrufen)

In diesem Abschnitt lesen wir die Analogdaten vom Potentiometer und drucken sie auf den seriellen Monitor.

```
inputValue = analogRead(analogPin); // Read the analog value from the potentiometer
Serial.print("Input: ");           // Print "Input: " to the serial monitor
Serial.println(inputValue);        // Print the raw input value to the serial
↪monitor
```

3. Kartierung und Skalierung (Umrechnung der Potentiometer-Daten)

Wir skalieren die Rohdaten vom Potentiometer, die im Bereich von 0-1023 liegen, in einen neuen Bereich von 0-255.

`map(value, fromLow, fromHigh, toLow, toHigh)` wird verwendet, um eine Zahl von einem Bereich in einen anderen umzurechnen. Wenn der Wert innerhalb des Bereichs von `fromLow` und `fromHigh` liegt, wird er in einen entsprechenden Wert innerhalb des Bereichs von `toLow` und `toHigh` umgewandelt, wobei die Proportionalität zwischen den beiden Bereichen erhalten bleibt.

In diesem Fall müssen wir, da der LED-Pin (Pin 9) einen Bereich von 0-255 hat, Werte im Bereich von 0-1023 auf dieselbe Skala von 0-255 abbilden.

```
outputValue = map(inputValue, 0, 1023, 0, 255); // Map the input value to a new
↪range
```

4. Steuerung der LED und serielle Ausgabe

Schließlich steuern wir die Helligkeit der LED basierend auf dem skalierten Wert und drucken den skalierten Wert zur Überwachung.

```
Serial.print("Output: "); // Print "Output: " to the serial
↪monitor
Serial.println(outputValue); // Print the scaled output value
↪to the serial monitor
analogWrite(ledPin, outputValue); // Control the LED brightness
↪based on the scaled value
delay(1000);
```

4.21 Joystick-Modul

4.21.1 Überblick

Ein Joystick ist ein Eingabegerät, das aus einem auf einer Basis schwenkbaren Stick besteht und seinen Winkel oder seine Richtung an das zu steuernde Gerät meldet. Joysticks werden häufig zur Steuerung von Videospielen und Robotern verwendet. Hier wird ein Joystick PS2 verwendet.

4.21.2 Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

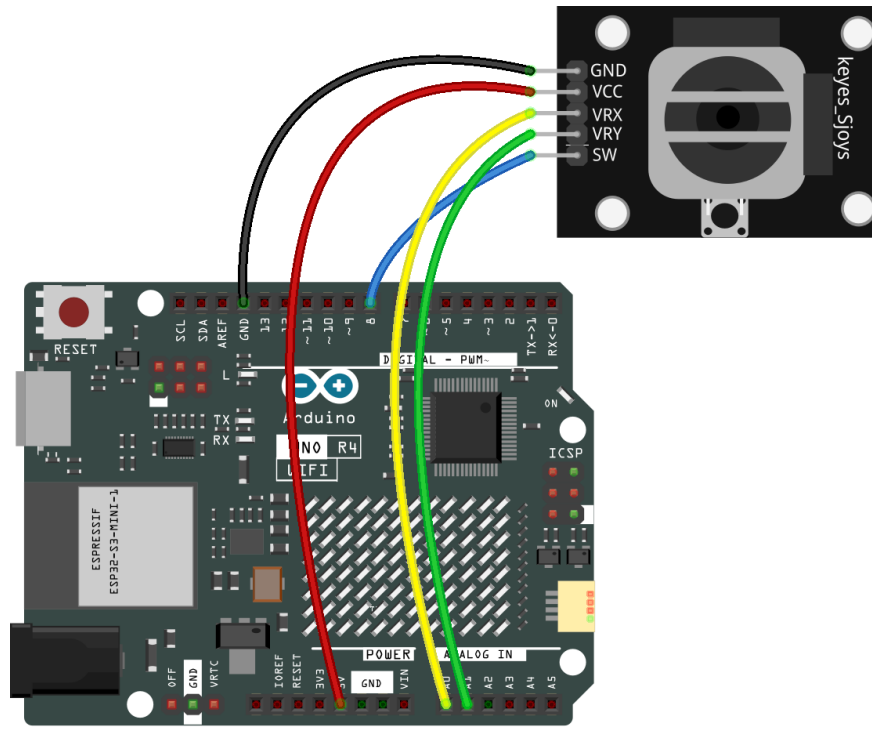
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Joystick-Modul</i>	

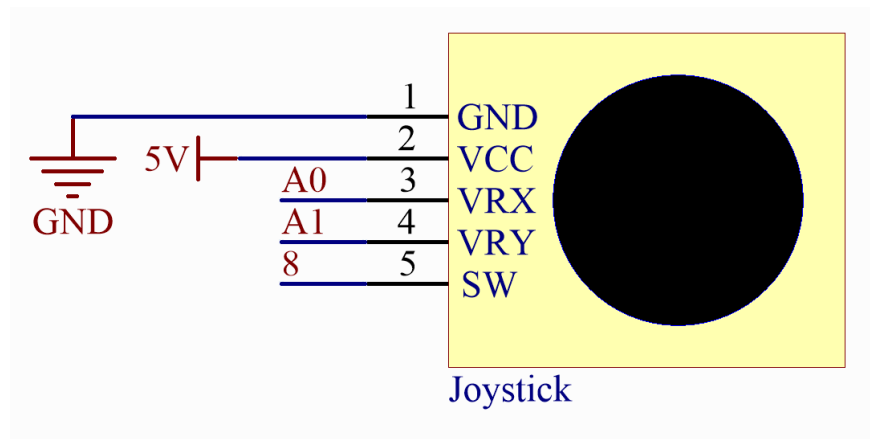
4.21.3 Verdrahtung



4.21.4 Schaltplan

Dieses Modul hat zwei analoge Ausgänge (entsprechend den X,Y-Biaxial-Versätzen).

In diesem Experiment verwenden wir das Uno-Board, um die Bewegungsrichtung des Joystick-Knopfes zu erkennen.



4.21.5 Code

Bemerkung:

- Sie können die Datei 20-joystick.ino direkt im Pfad elite-explorer-kit-main\basic_project\20-joystick öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Nun, wenn Sie den Rocker bewegen, ändern sich die Koordinaten der X- und Y-Achse auf dem Serial Monitor entsprechend. Wenn Sie den Knopf drücken, wird auch die Koordinate Z=0 angezeigt.

4.21.6 Code-Analyse

Der Code verwendet den seriellen Monitor, um den Wert der VRX- und VRY-Pins des Joystick ps2 auszugeben.

```
void loop()
{
  Serial.print("X: ");
  Serial.print(analogRead(xPin), DEC); // print the value of VRX in DEC
  Serial.print("|Y: ");
  Serial.print(analogRead(yPin), DEC); // print the value of VRY in DEC
  Serial.print("|Z: ");
  Serial.println(digitalRead(swPin)); // print the value of SW
  delay(50);
}
```

4.22 Tastenfeld

4.22.1 Überblick

In dieser Lektion lernen Sie, wie man ein Tastenfeld verwendet. Tastenfelder können in verschiedene Arten von Geräten eingesetzt werden, einschließlich Mobiltelefone, Faxgeräte, Mikrowellenherde und so weiter. Sie werden häufig für Benutzereingaben verwendet.

4.22.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

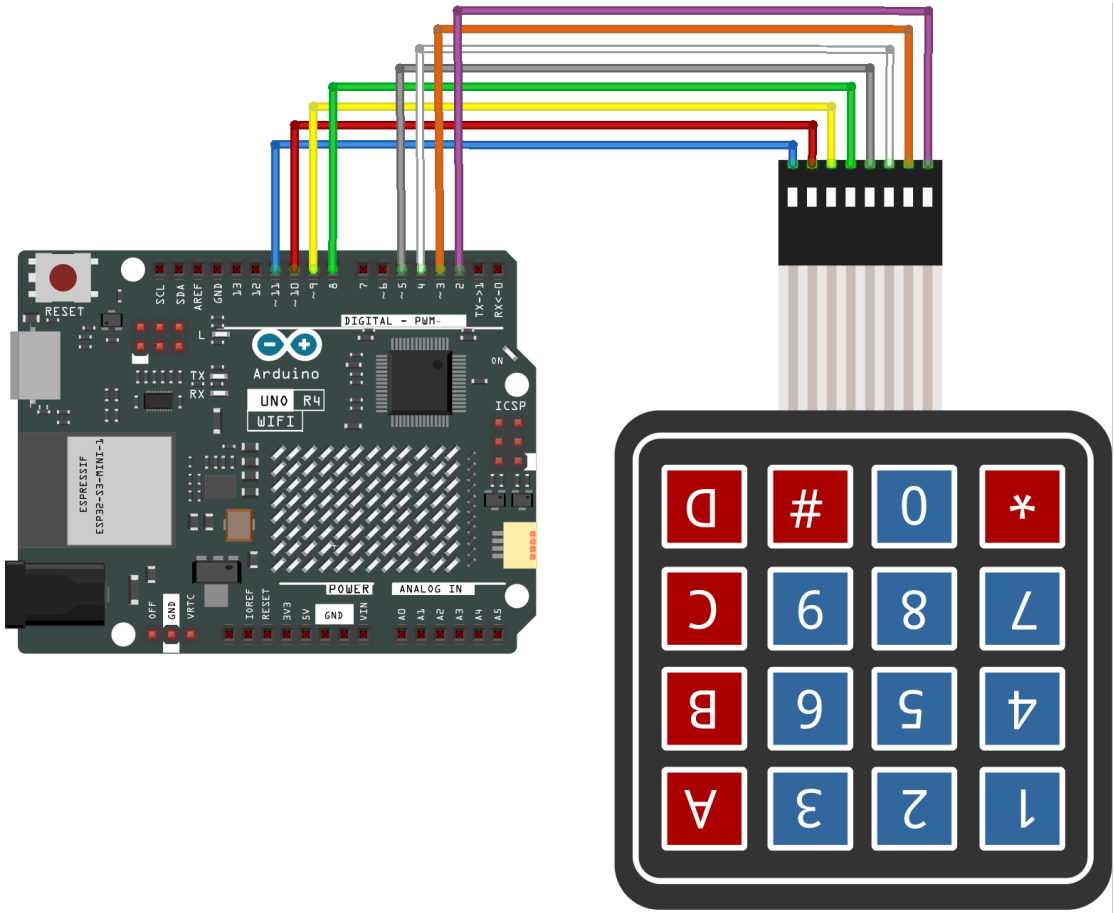
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

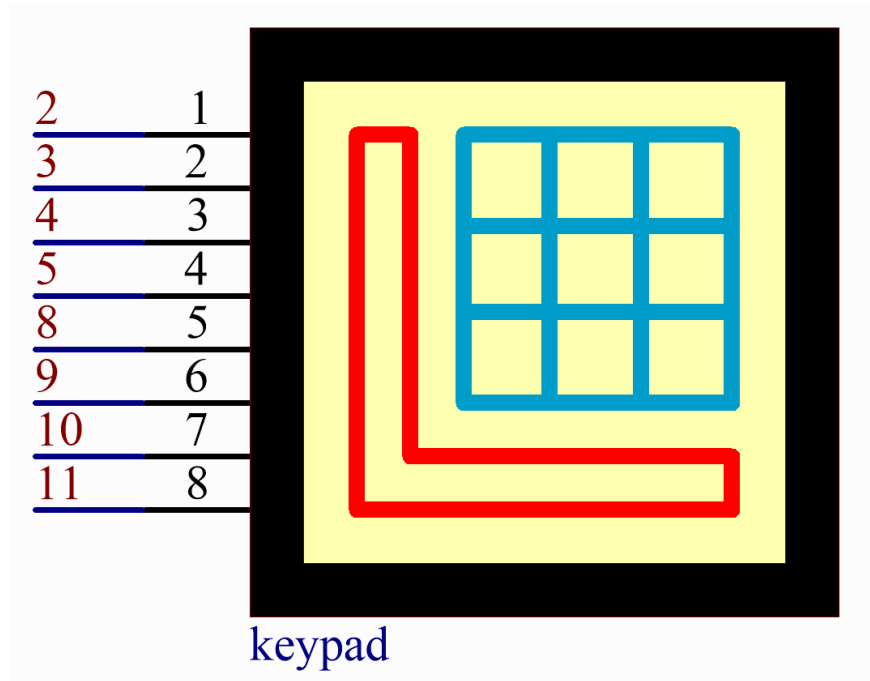
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Jumperkabel	
Tastenfeld	

4.22.3 Verdrahtung



4.22.4 Schaltplan



4.22.5 Code

Bemerkung:

- Sie können die Datei 21-keypad.ino direkt im Pfad elite-explorer-kit-main\basic_project\21-keypad öffnen.
 - Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen nach „**Adafruit Keypad**“ und installieren es.
-

Nachdem die Codes auf das UNO-Board hochgeladen wurden, können Sie auf dem seriellen Monitor den Wert der aktuell auf dem Tastenfeld gedrückten Taste sehen.

4.22.6 Code-Analyse

1. Einbinden der Bibliothek

Wir beginnen mit dem Einbinden der Adafruit_Keypad Bibliothek, die es uns ermöglicht, einfach mit dem Tastenfeld zu interagieren.

```
#include "Adafruit_Keypad.h"
```

Bemerkung:

- Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen nach „**Adafruit Keypad**“ und installieren es.
-

2. Konfiguration des Tastenfelds

```
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};
byte rowPins[ROWS] = { 2, 3, 4, 5 };
byte colPins[COLS] = { 8, 9, 10, 11 };
```

- Die Konstanten ROWS und COLS definieren die Abmessungen des Tastenfelds.
- keys ist ein 2D-Array, das die Beschriftung jeder Taste auf dem Tastenfeld speichert.
- rowPins und colPins sind Arrays, die die mit den Reihen und Spalten des Tastenfelds verbundenen Arduino-Pins speichern.

3. Initialisierung des Tastenfelds

Erstellen Sie eine Instanz von Adafruit_Keypad namens myKeypad und initialisieren Sie diese.

```
Adafruit_Keypad myKeypad = Adafruit_Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
↪ COLS);
```

4. setup()-Funktion

Initialisieren Sie die serielle Kommunikation und das benutzerdefinierte Tastenfeld.

```
void setup() {
  Serial.begin(9600);
  myKeypad.begin();
}
```

5. Hauptloop

Überprüfen Sie auf Tastenereignisse und zeigen Sie diese im seriellen Monitor an.

```
void loop() {
  myKeypad.tick();
  while (myKeypad.available()) {
    keypadEvent e = myKeypad.read();
    Serial.print((char)e.bit.KEY);
    if (e.bit.EVENT == KEY_JUST_PRESSED) Serial.println(" pressed");
    else if (e.bit.EVENT == KEY_JUST_RELEASED) Serial.println(" released");
  }
  delay(10);
}
```

4.23 Infrarot-Empfänger

4.23.1 Überblick

Ein Infrarot-Empfänger ist eine Komponente, die Infrarotsignale empfängt und unabhängig Infrarotstrahlen empfangen und Signale ausgeben kann, die mit TTL-Pegel kompatibel sind. Er ähnelt in der Größe einem normalen, in Kunststoff verpackten Transistor und eignet sich für alle Arten von Infrarot-Fernbedienungen und Infrarot-Übertragungen.

4.23.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

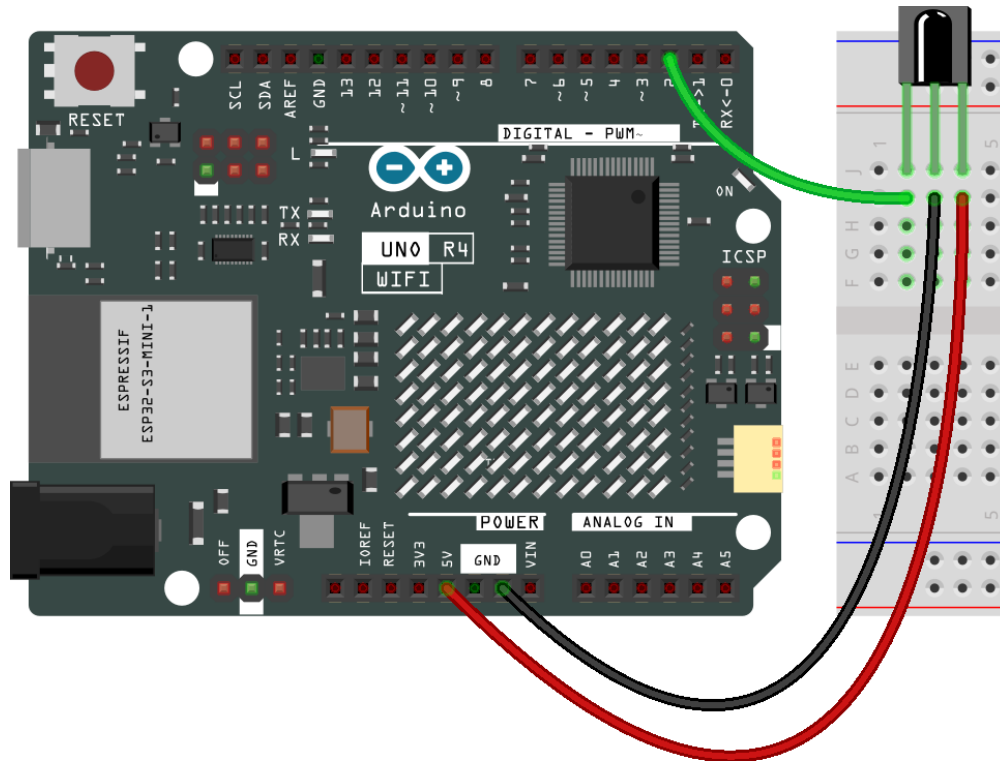
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

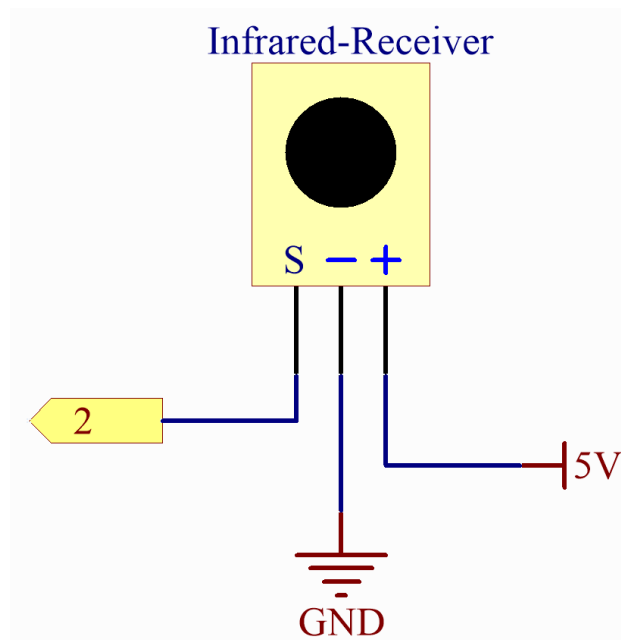
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Infrarotempfänger</i>	

4.23.3 Verdrahtung



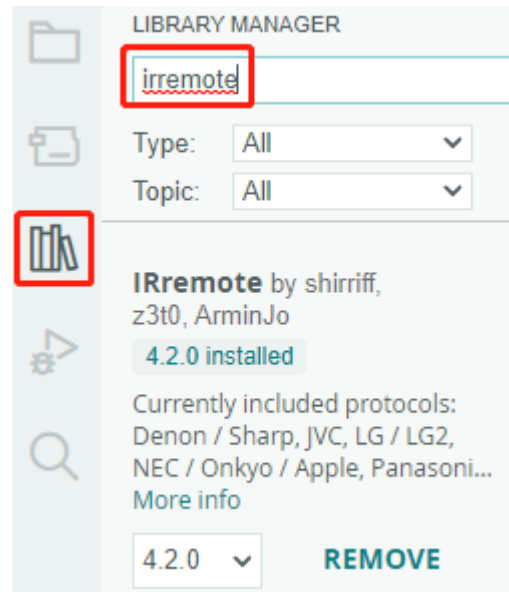
4.23.4 Schaltplan



4.23.5 Code

Bemerkung:

- Sie können die Datei `22-ir_receiver.ino` direkt im Pfad `elite-explorer-kit-main\basic_project\22-ir_receiver` öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-
- Hier wird die IRremote Bibliothek verwendet, die Sie über den **Library Manager** installieren können.



Bemerkung:

- An der Rückseite der Fernbedienung befindet sich ein transparentes Kunststoffteil, um die Stromzufuhr zu unterbrechen. Ziehen Sie es heraus, bevor Sie die Fernbedienung benutzen.

4.23.6 Code-Analyse

Dieser Code ist für die Verwendung mit einer Infrarot (IR) Fernbedienung und der IRremote Bibliothek konzipiert. Hier ist die Aufschlüsselung:

1. Einbinden der Bibliothek und Definieren von Konstanten. Zunächst wird die IRremote Bibliothek eingebunden, und die Pinnummer für den IR-Empfänger wird als 2 definiert.

```
#include <IRremote.h>
const int IR_RECEIVE_PIN = 2;
```

2. Initialisiert die serielle Kommunikation mit einer Baudrate von 9600. Initialisiert den IR-Empfänger am angegebenen Pin (IR_RECEIVE_PIN) und aktiviert die LED-Rückmeldung (falls zutreffend).

```
void setup() {
    Serial.begin(9600);
```

// Start serial

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

↪ communication at 9600 baud rate
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // Start the IR receiver
}

```

3. Die Schleife läuft kontinuierlich, um eingehende IR-Fernbedienungssignale zu verarbeiten.

```

void loop() {
  // Check if there is any incoming IR signal
  if (IrReceiver.decode()) {
    // IrReceiver.printIRResultShort(&Serial);           // Print the
↪ received data in one line
    // Serial.println(IrReceiver.decodedIRData.command, HEX); // Print the command
↪ in hexadecimal format
    Serial.println(decodeKeyValue(IrReceiver.decodedIRData.command)); // Map and
↪ print the decoded IR signal to corresponding key value

    IrReceiver.resume(); // Enable receiving of the next value
  }
}

```

- Überprüft, ob ein IR-Signal empfangen und erfolgreich decodiert wurde.
- Decodiert den IR-Befehl und speichert ihn in decodedValue mit einer benutzerdefinierten Funktion decodeKeyValue().
- Druckt den decodierten IR-Wert auf den seriellen Monitor.
- Setzt den IR-Signalempfang für das nächste Signal fort.

4. Hilfsfunktion, um empfangene IR-Signale den entsprechenden Tasten zuzuordnen



```
// Function to map received IR signals to corresponding keys
String decodeKeyValue(long result) {
    // Each case corresponds to a specific IR command
    switch (result) {
        case 0x16:
            return "0";
        case 0xC:
            return "1";
        case 0x18:
            return "2";
        case 0x5E:
            return "3";
        case 0x8:
            return "4";
        case 0x1C:
            return "5";
        case 0x5A:
            return "6";
        case 0x42:
            return "7";
        case 0x52:
            return "8";
        case 0x4A:
            return "9";
        case 0x9:
            return "+";
        case 0x15:
            return "-";
        case 0x7:
            return "EQ";
        case 0xD:
            return "U/SD";
        case 0x19:
            return "CYCLE";
        case 0x44:
            return "PLAY/PAUSE";
        case 0x43:
            return "FORWARD";
        case 0x40:
            return "BACKWARD";
        case 0x45:
            return "POWER";
        case 0x47:
            return "MUTE";
        case 0x46:
            return "MODE";
        case 0x0:
            return "ERROR";
        default:
            return "ERROR";
    }
}
```

4.24 MPR121

4.24.1 Überblick

In dieser Lektion lernen Sie, wie man MPR121 verwendet. Es ist eine gute Wahl, wenn Sie viele Berührungsschalter zu Ihrem Projekt hinzufügen möchten. Die Elektrode des MPR121 kann mit einem Leiter verlängert werden. Wenn Sie einen Draht an eine Banane anschließen, können Sie die Banane in einen Berührungsschalter verwandeln und so Projekte wie ein Fruchtpiano realisieren.

4.24.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

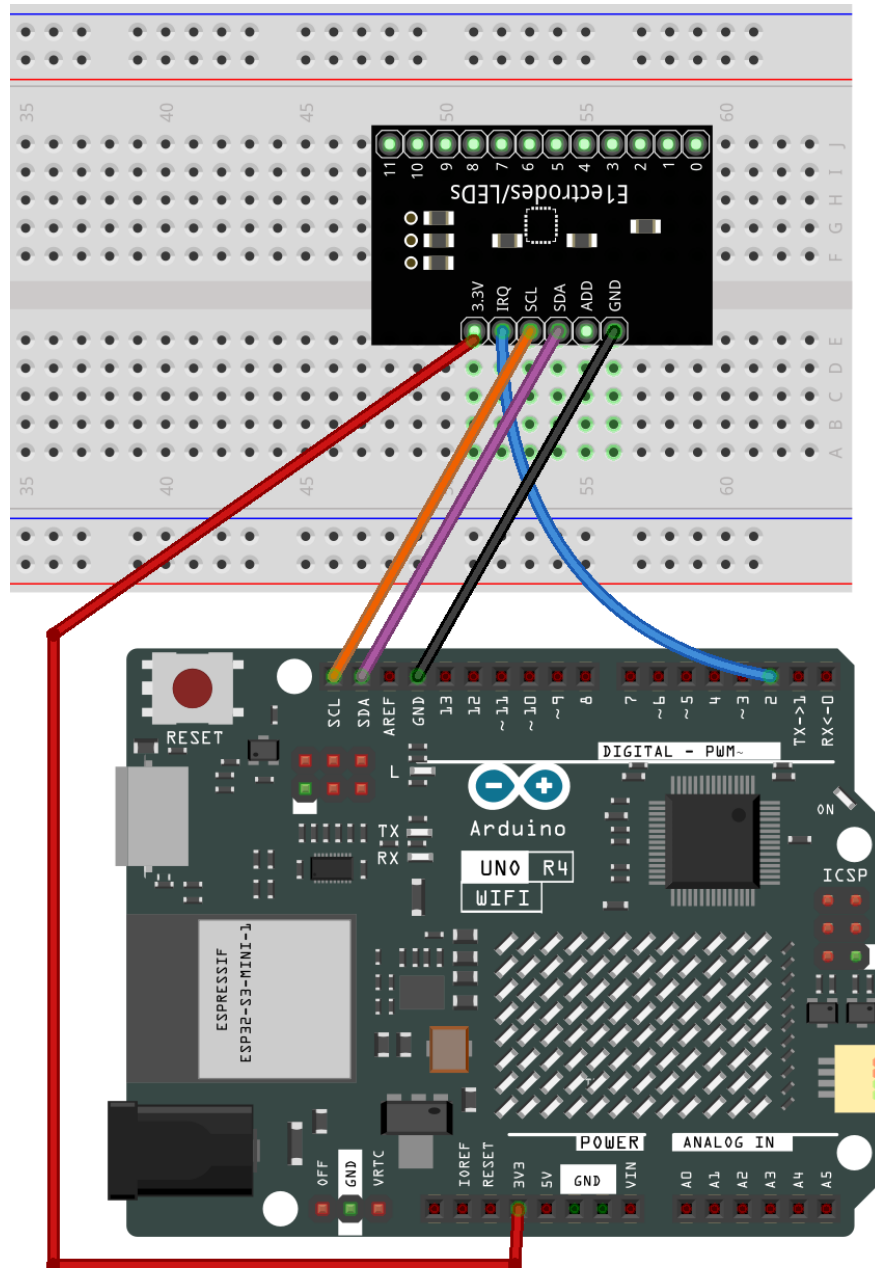
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>MPR121</i>	-

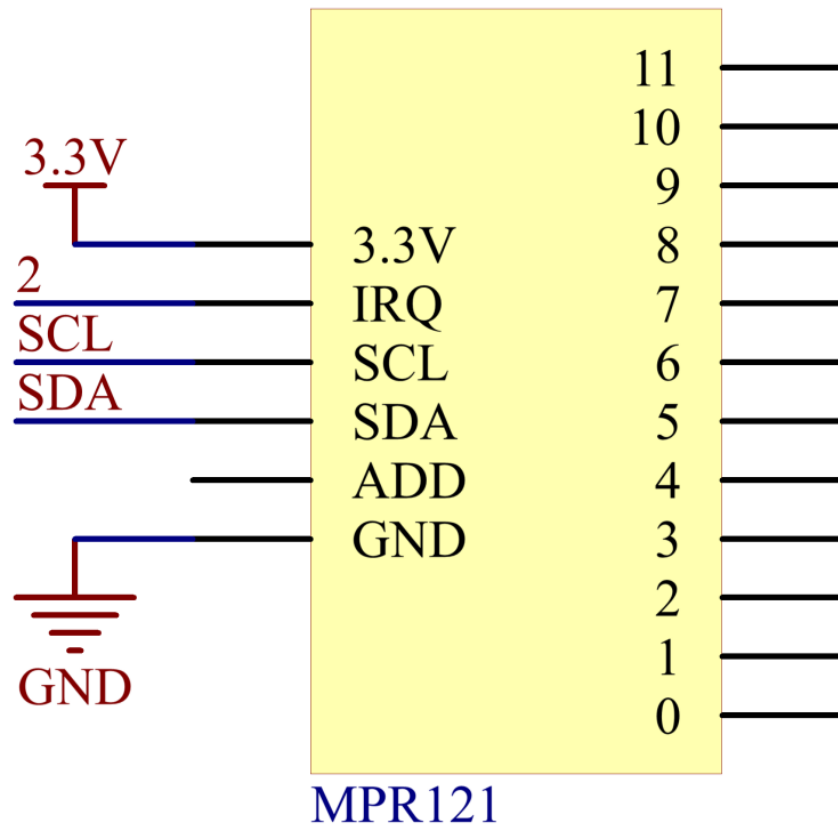
4.24.3 Verdrahtung

In diesem Beispiel setzen wir MPR121 in das Breadboard ein. Verbinden Sie GND von MPR121 mit GND, 3.3V mit 3V3, IRQ mit dem digitalen Pin 2, SCL mit dem Pin SCL(A5) und SDA mit dem Pin SDA(A4). Es gibt 12 Elektroden für die Berührungserkennung.

Bemerkung: MPR121 wird mit 3.3V betrieben, nicht mit 5V.



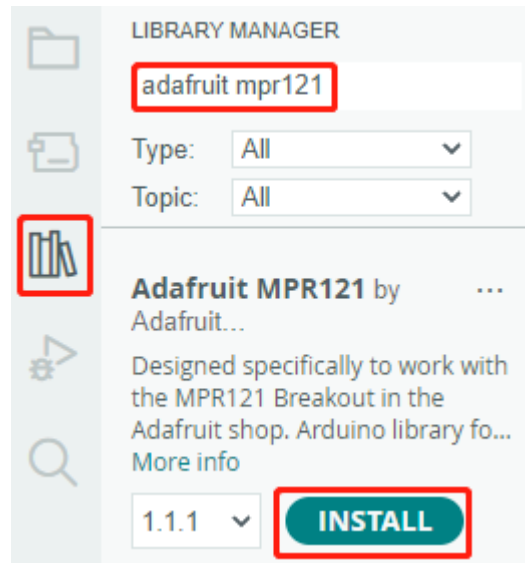
4.24.4 Schaltplan



4.24.5 Code

Bemerkung:

- Sie können die Datei 23-mpr121.ino direkt im Pfad elite-explorer-kit-main\basic_project\23-mpr121 öffnen.
- Hier wird die Adafruit MPR121 Bibliothek verwendet, die Sie über den **Library Manager** installieren können.



Nachdem der Code auf das UNO-Board hochgeladen wurde, wird der Berührungszustand der MPR121-Pins „1“ und „0“ in einem 12-Bit-Boolean-Array erfasst. Dieses Array wird dann auf dem seriellen Monitor ausgegeben.

4.24.6 Code-Analyse

Dieser Code ermöglicht die Kommunikation und Bedienung des MPR121-Berührungssensors. Er kann den Status von Berührungselektroden erkennen und Informationen über berührte oder freigegebene Elektroden auf der seriellen Schnittstelle ausgeben. Wenn detaillierte Sensordaten erforderlich sind, kann der entsprechende Code auskommentiert werden.

Hier eine Analyse des Codes:

1. Bibliotheken importieren:

```
#include <Wire.h>
#include "Adafruit_MPR121.h"
```

- `Wire.h`: Wird für die I2C-Kommunikation verwendet.
- `Adafruit_MPR121.h`: Adafruits MPR121-Bibliothek zur Bedienung des MPR121-Berührungssensors.

2. Die Makro `_BV` definieren:

```
#ifndef _BV
#define _BV(bit) (1 << (bit))
#endif
```

`_BV(bit)` definiert ein Makro, das einen gegebenen Bit in den entsprechenden Binärwert umwandelt, ähnlich wie `1 << bit`.

3. Instanz der Klasse `Adafruit_MPR121` initialisieren:

```
Adafruit_MPR121 cap = Adafruit_MPR121();
```

Erstellen Sie eine Instanz der Klasse `Adafruit_MPR121` namens `cap`. Das Objekt `cap` wird verwendet, um mit dem MPR121-Berührungssensor zu kommunizieren und ihn zu bedienen.

4. Funktion setup():

Initialisieren Sie die serielle Kommunikation mit einer Baudrate von 9600. Dann initialisieren Sie den MPR121-Berührungssensor mit der Standard-I2C-Adresse 0x5A. Falls die Initialisierung fehlschlägt, geben Sie eine Fehlermeldung aus und treten Sie in eine Endlosschleife ein.

```
void setup() {
    Serial.begin(9600);

    while (!Serial) { // needed to keep leonardo/micro from starting too fast!
        delay(10);
    }

    Serial.println("Adafruit MPR121 Capacitive Touch sensor test");

    // Default address is 0x5A, if tied to 3.3V its 0x5B
    // If tied to SDA its 0x5C and if SCL then 0x5D
    if (!cap.begin(0x5A)) {
        Serial.println("MPR121 not found, check wiring?");
        while (1);
    }
    Serial.println("MPR121 found!");
}
```

5. Funktion loop():

- Ermitteln Sie den aktuellen Berührungsstatus, der als 16-Bit-Integer zurückgegeben wird.

```
curr_touched = cap.touched();
```

- Durchlaufen Sie den Status der 12 Elektroden (nummeriert von 0 bis 11).

```
for (uint8_t i=0; i<12; i++) {
    // it if *is* touched and *wasnt* touched before, alert!
    if ((curr_touched & _BV(i)) && !(last_touched & _BV(i)) ) {
        Serial.print(i); Serial.println(" touched");
    }
    // if it *was* touched and now *isnt*, alert!
    if (!(curr_touched & _BV(i)) && (last_touched & _BV(i)) ) {
        Serial.print(i); Serial.println(" released");
    }
}
```

- Wenn eine Elektrode berührt wird und vorher nicht berührt wurde, drucken Sie „x berührt“, wobei x die Nummer der Elektrode ist.
- Wenn eine Elektrode vorher berührt wurde, aber jetzt nicht mehr berührt wird, drucken Sie „x freigegeben“.
- Aktualisieren Sie last_touched, um den aktuellen Berührungsstatus für den Vergleich in der nächsten Iteration zu speichern.

```
last_touched = curr_touched;
```

- Debugging-Informationen (Optional):

[illegible]

Aktuator

4.25 Motor

4.25.1 Überblick

In dieser Lektion lernen Sie, wie man einen Motor verwendet, dessen Arbeitsprinzip darauf beruht, dass die stromdurchflossene Spule im Magnetfeld zur Rotation gezwungen wird und somit der Rotor des Motors entsprechend rotiert, wodurch das Ritzelrad das Schwungrad des Motors antreibt.

4.25.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

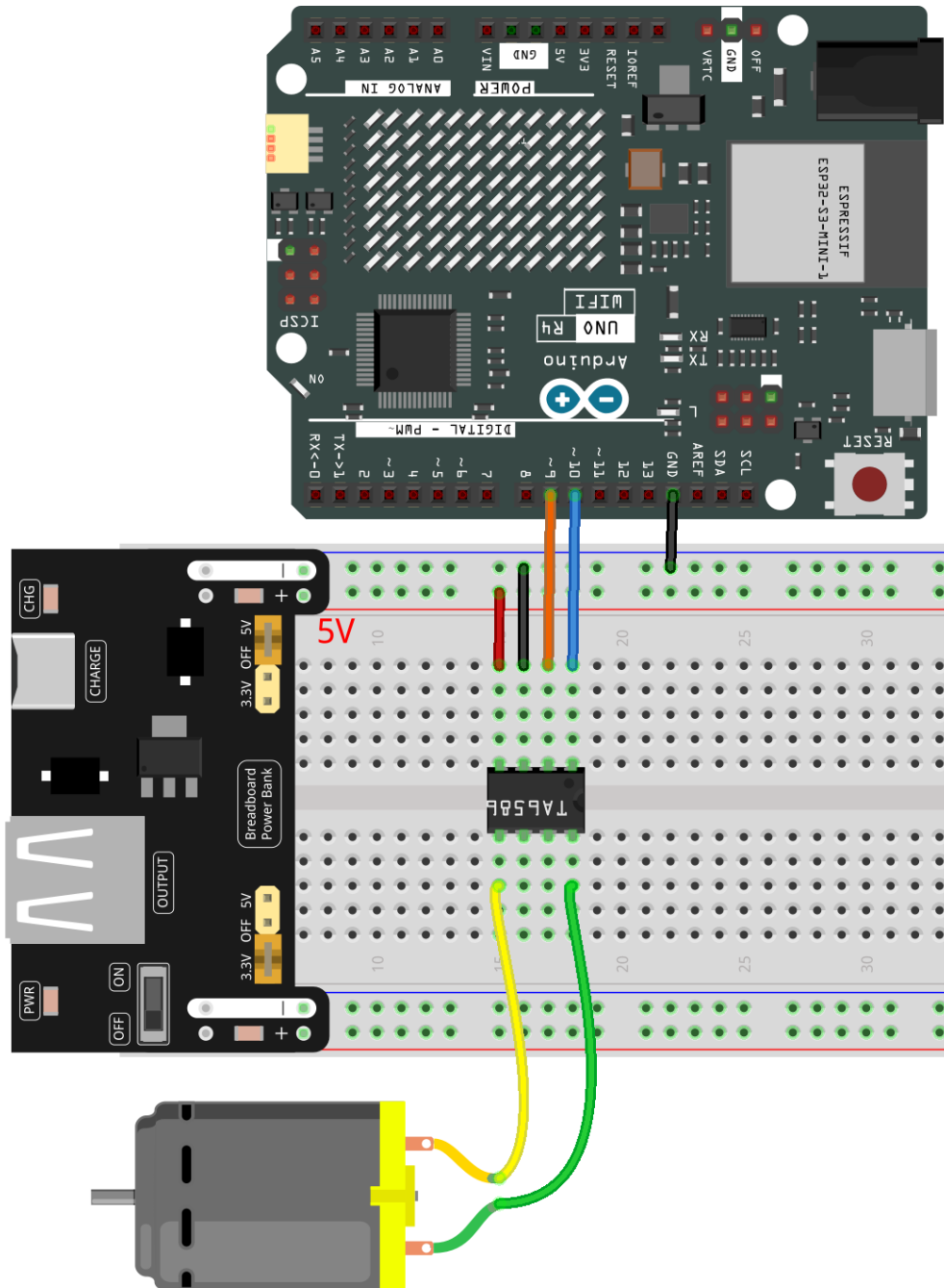
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

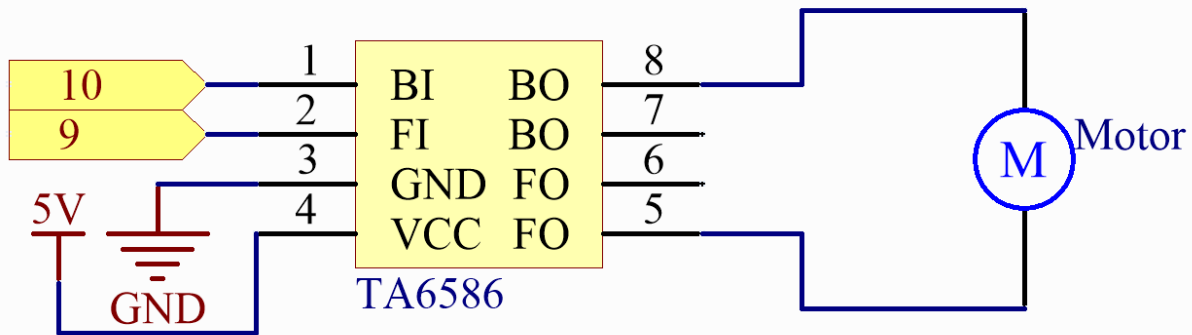
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>TA6586 - Motorsteuerungs-Chip</i>	-
<i>Gleichstrommotor</i>	
<i>Stromversorgungsmodul</i>	-

4.25.3 Verdrahtung

In diesem Beispiel verwenden wir das Power Supply Module, um die Anode und Kathode des Breadboards zu versorgen.



4.25.4 Schaltplan



4.25.5 Code

Bemerkung:

- Sie können die Datei 24-motor.ino direkt im Pfad elite-explorer-kit-main\basic_project\24-motor öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem der Code auf das UNO-Board hochgeladen wurde, können Sie die Drehrichtung des Motors wählen, indem Sie „A“ oder „B“ im seriellen Monitor eingeben.

4.25.6 Code-Analyse

Der Motor kann angetrieben werden, indem man eine Spannungsdifferenz zwischen den Kupferplatten auf beiden Seiten des Motors herstellt. Daher müssen Sie nur 0 für die Spannung einer Seite der Kupferplatte und 5V für die andere Seite schreiben. Modifizieren Sie den geschriebenen analogen Signalwert, um die Richtung und Geschwindigkeit anzupassen.

```
// Function to rotate the motor clockwise
void clockwise(int Speed) {
  analogWrite(motorBI, 0);
  analogWrite(motorFI, Speed);
}

// Function to rotate the motor anticlockwise
void anticlockwise(int Speed) {
  analogWrite(motorBI, Speed);
  analogWrite(motorFI, 0);
}
```

In diesem Beispiel wird Serial.Read() verwendet, um die Richtung des Motors zu steuern.

Wenn Sie „A“ im seriellen Monitor eingeben, wird die Funktion clockwise (255) aufgerufen, damit sich der Motor mit der Geschwindigkeit von 255 dreht. Geben Sie „B“ ein, und der Motor dreht sich in die entgegengesetzte Richtung.

```

void loop() {
  // Check if there is available data on the serial port
  if (Serial.available() > 0) {
    int incomingByte = Serial.read(); // Read incoming data

    // Determine motor direction based on user input
    switch (incomingByte) {
      case 'A':
        clockwise(255); // Rotate motor clockwise
        Serial.println("The motor rotates clockwise.");
        break;
      case 'B':
        anticlockwise(255); // Rotate motor anticlockwise
        Serial.println("The motor rotates anticlockwise.");
        break;
    }
  }

  delay(3000); // Wait for 3 seconds
  stopMotor(); // Stop the motor
}

```

4.26 Wasserpumpe

4.26.1 Überblick

Die Wasserpumpe ist auch ein Motor, der die mechanische Energie des Motors oder anderer externer Energie durch eine spezielle Struktur umwandelt, um die Flüssigkeit zu transportieren.

4.26.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

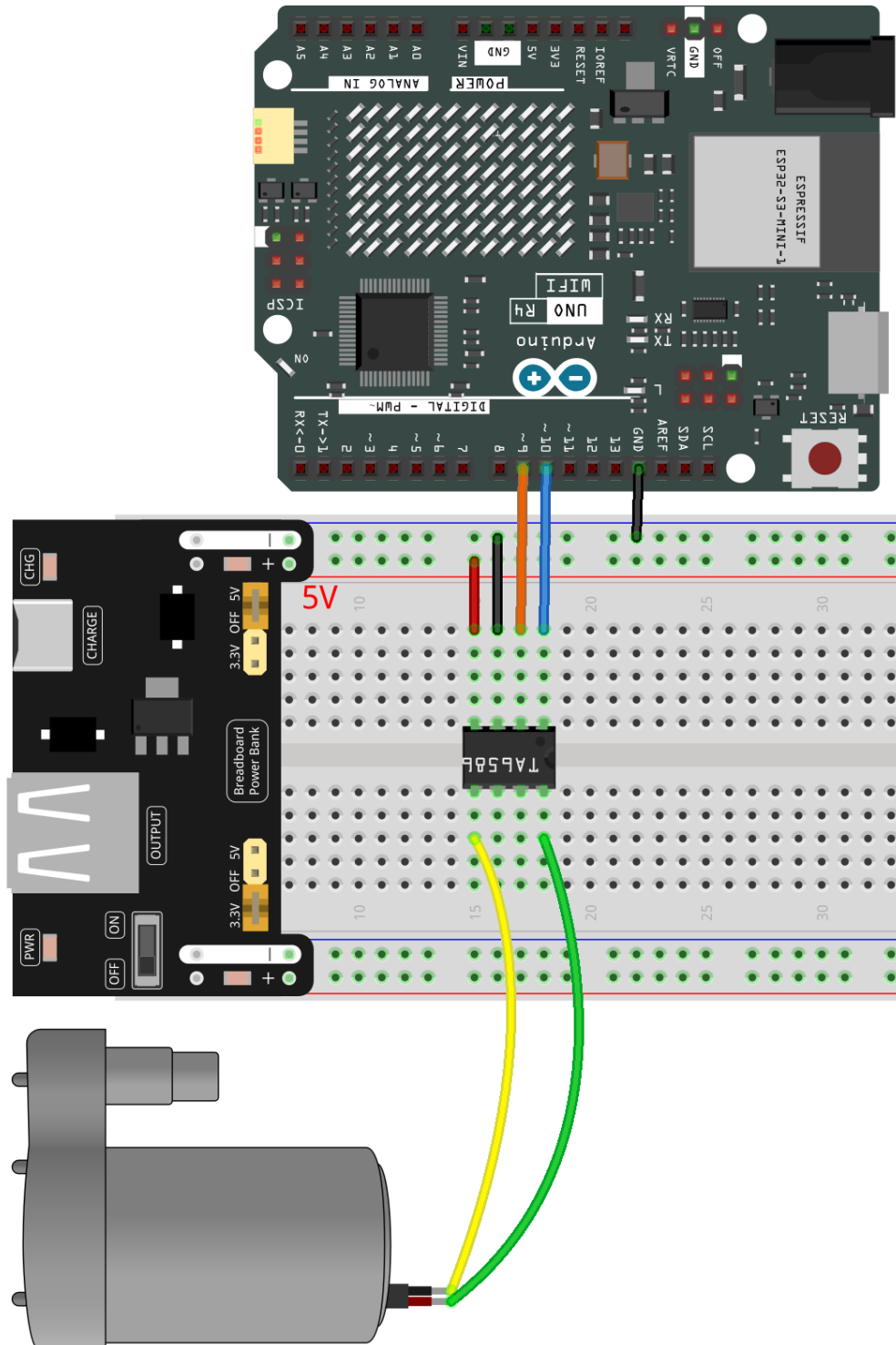
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

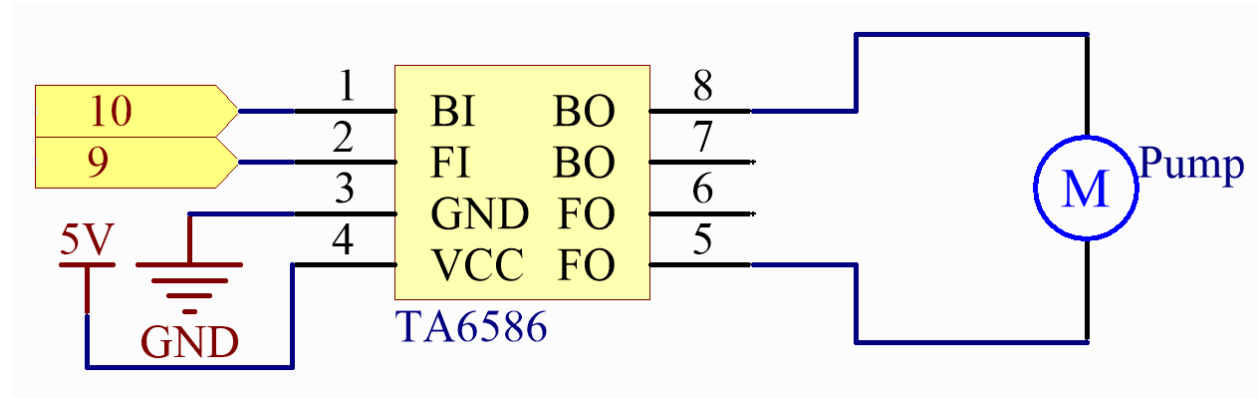
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>DC-Wasserpumpe</i>	
<i>TA6586 - Motorsteuerungs-Chip</i>	-
<i>Stromversorgungsmodul</i>	-

4.26.3 Verdrahtung



4.26.4 Schaltplan



4.26.5 Code

Bemerkung:

- Sie können die Datei 25-pump.ino direkt im Pfad elite-explorer-kit-main\basic_project\25-pump öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Befestigen Sie den Schlauch an der Pumpe und positionieren Sie sie im Becken. Sobald der Code erfolgreich hochgeladen wurde, schaltet sich die Wasserpumpe ein und bleibt fünf Sekunden lang aktiv. Bitte achten Sie bei diesem Experiment darauf, dass der Stromkreis vom Wasser ferngehalten wird, um mögliche Kurzschlüsse zu vermeiden.

4.26.6 Code-Analyse

Der Motor kann betrieben werden, indem man eine Spannungsdifferenz zwischen den Kupferplatten auf beiden Seiten des Motors herstellt.

```
digitalWrite(motorBI, HIGH);
digitalWrite(motorFI, LOW);
```

4.27 Schrittmotor

4.27.1 Überblick

In dieser Lektion lernen Sie, wie man Schrittmotoren steuert, speziell das Modell 28BYJ-48, unter Verwendung eines ULN2003-Treibers und eines Arduino Uno R4. Schrittmotoren werden in einer Vielzahl von Anwendungen wie 3D-Druckern, CNC-Maschinen, Robotik und sogar in gängigen Haushaltsgeräten verwendet. Ihre präzise Steuerung ermöglicht komplexe Bewegungen, was sie ideal für Projekte macht, die eine hohe Positionsgenauigkeit erfordern.

In diesem Projekt konfigurieren wir den 28BYJ-48-Schrittmotor so, dass er sich in beide Richtungen – im Uhrzeigersinn und gegen den Uhrzeigersinn – mit verschiedenen Geschwindigkeiten dreht. Solche Schrittmotoren werden oft in automatisierten Systemen verwendet, um Objekte zu drehen oder Mechanismen anzutreiben, die eine präzise

Steuerung erfordern. Beispielsweise können sie in automatischen Vorhängen verwendet werden, die sich zu bestimmten Zeiten oder unter bestimmten Bedingungen öffnen oder schließen. Wenn Sie verstehen, wie man die Drehung und Geschwindigkeit eines Schrittmotors steuert, sind Sie auf dem besten Weg, sie in Ihre eigenen elektronischen Projekte zu integrieren.

4.27.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

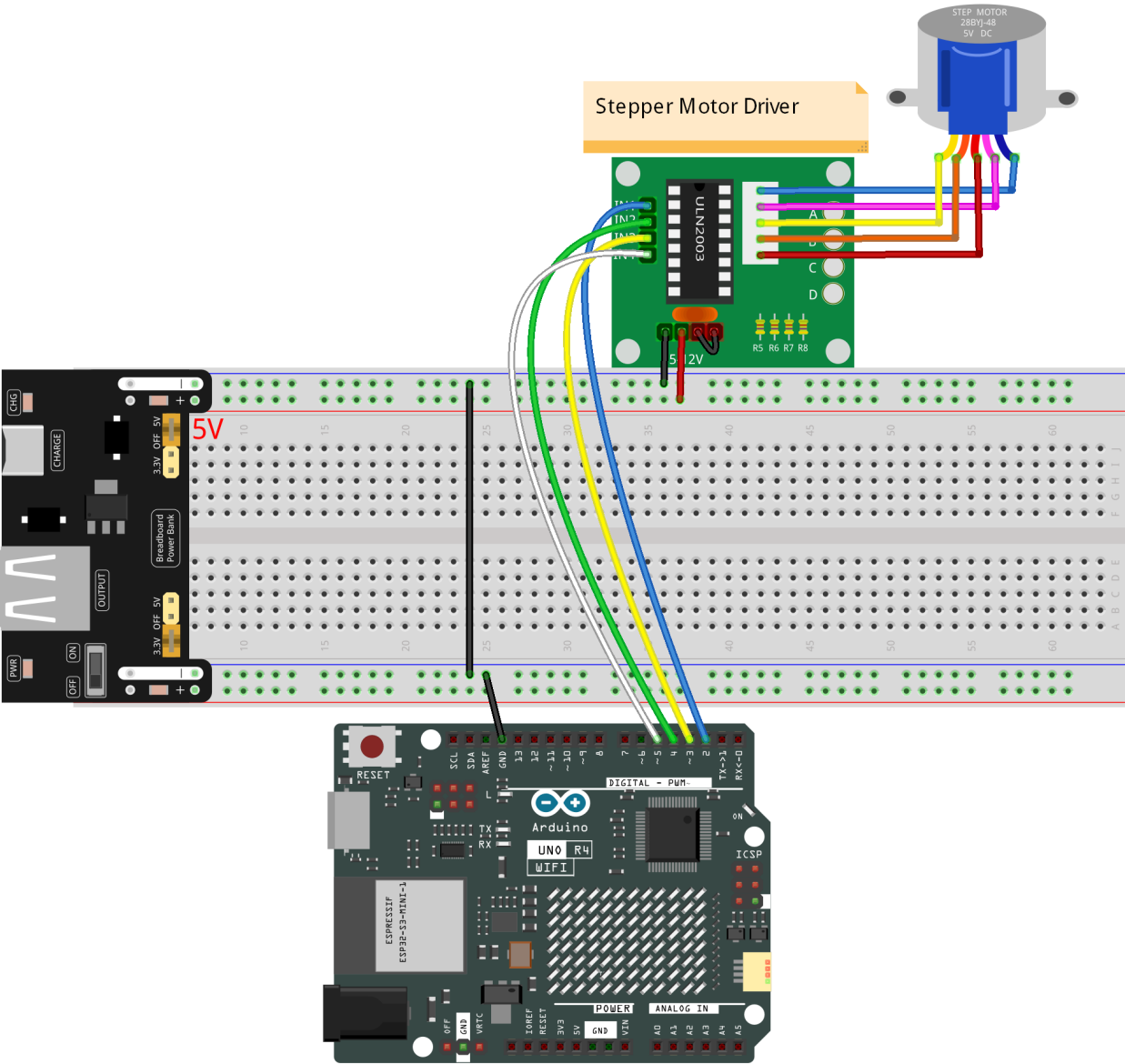
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Schrittmotor</i>	
<i>Stromversorgungsmodul</i>	-

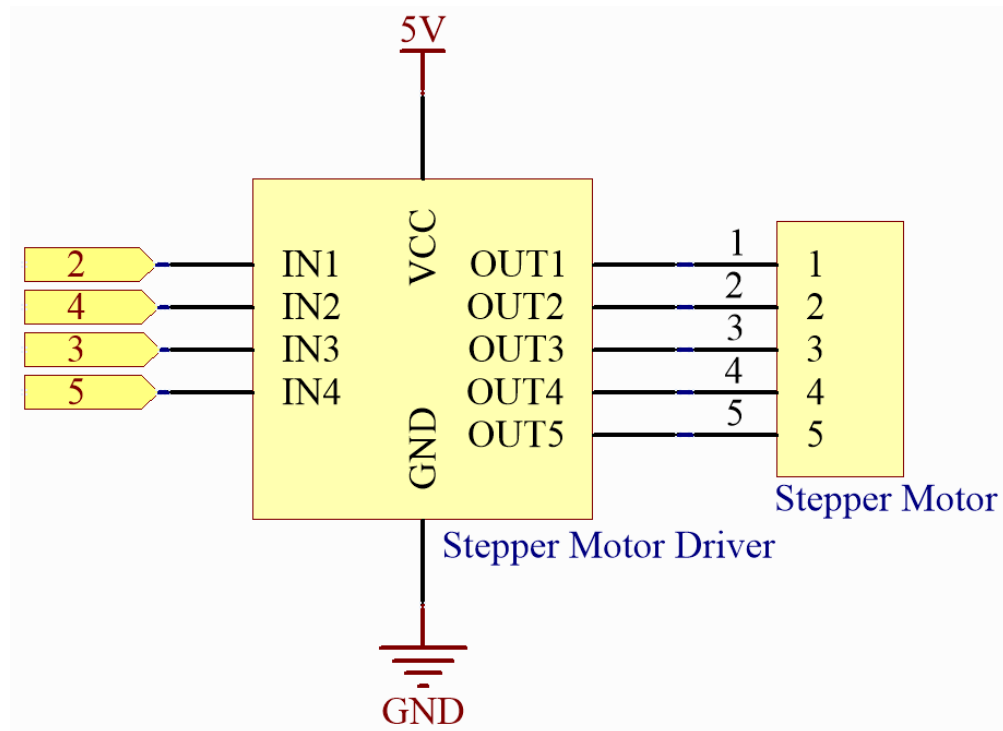
4.27.3 Verdrahtung

Warnung: Aufgrund des hohen Stromverbrauchs des Schrittmotors ist es ratsam, eine externe 5V-Stromversorgung zu verwenden, anstatt sich auf den Arduino zu verlassen.

Obwohl es möglich ist, den Schrittmotor direkt vom Arduino zu betreiben, wird dies nicht empfohlen, da es elektrisches Rauschen auf dessen Stromversorgungsleitungen verursachen kann, was möglicherweise zu Schäden am Arduino führt.



4.27.4 Schaltplan



4.27.5 Code

Bemerkung:

- Sie können die Datei `26-stepper_motor.ino` direkt im Pfad `elite-explorer-kit-main\basic_project\26-stepper_motor.rst` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Nachdem der Code auf das Arduino Uno Board hochgeladen wurde, beginnt der 28BYJ-48 Schrittmotor sich zu drehen, angetrieben durch den ULN2003 Treiber. Zunächst dreht sich der Motor im Uhrzeigersinn mit einer Geschwindigkeit von 5 Umdrehungen pro Minute (RPM) für eine vollständige Umdrehung. Nachdem die Drehung im Uhrzeigersinn abgeschlossen ist, hält der Motor für 1 Sekunde an.

Anschließend dreht sich der Motor in die entgegengesetzte Richtung, gegen den Uhrzeigersinn, mit einer erhöhten Geschwindigkeit von 15 RPM für eine weitere vollständige Umdrehung. Wieder hält der Motor nach der Drehung gegen den Uhrzeigersinn für 1 Sekunde an. Der Zyklus aus Drehung und Pause setzt sich unendlich fort, solange das Arduino mit Strom versorgt wird.

4.27.6 Code-Analyse

1. Initialisierung des Schrittmotors

```
#include <Stepper.h> // Include the Stepper library

#define STEPS 2038 // Define the number of steps per revolution
↳ for the motor
Stepper stepper(STEPS, 2, 3, 4, 5); // Initialize stepper object and set pin
↳ connections (IN1, IN2, IN3, IN4)
```

Binden Sie die Kopffdatei `Stepper.h` ein, setzen Sie die Schritte auf 2038 und initialisieren Sie den Schrittmotor mit der Funktion `stepper()`.

STEPS: Die Anzahl der Schritte in einer Umdrehung Ihres Motors. Für diesen Schrittmotor beträgt dieser Wert 2038.

Stepper(steps, pin1, pin2, pin3, pin4): Diese Funktion erstellt eine neue Instanz der Stepper-Klasse, die einen bestimmten Schrittmotor repräsentiert, der an Ihrem Arduino-Board angeschlossen ist. Die Pins `pin1`, `pin2`, `pin3` und `pin4` entsprechen den IN1-, IN2-, IN3- und IN4-Pins am ULN2003-Treiber.

2. loop() Funktion

```
void loop() {
  // Rotate clockwise at 5 RPM
  stepper.setSpeed(5);
  stepper.step(STEPS); // Rotate one full revolution clockwise
  delay(1000); // Wait for 1 second

  // Rotate counter-clockwise at 15 RPM
  stepper.setSpeed(15);
  stepper.step(-STEPS); // Rotate one full revolution counter-clockwise
  delay(1000); // Wait for 1 second
}
```

Das Hauptprogramm dreht den Schrittmotor kontinuierlich, einmal eine volle Drehung im Uhrzeigersinn mit 5 U/min und dann einmal eine volle Drehung gegen den Uhrzeigersinn mit 15 U/min.

- **setSpeed(rpms):** Legt die Motorgeschwindigkeit in Umdrehungen pro Minute (U/min) fest. Diese Funktion bringt den Motor nicht zum Drehen, sondern legt nur die Geschwindigkeit fest, mit der er sich dreht, wenn Sie `step()` aufrufen.
 - **rpms:** die Geschwindigkeit, mit der sich der Motor in Umdrehungen pro Minute drehen soll – eine positive Zahl (long)
- **step(steps):** Diese Funktion dreht den Motor um eine bestimmte Anzahl von Schritten, wobei die in der letzten `setSpeed()`-Aufruf festgelegte Geschwindigkeit verwendet wird. Es ist wichtig zu beachten, dass diese Funktion blockierend arbeitet, das heißt, sie wartet, bis der Motor seine Bewegung abgeschlossen hat, bevor die Steuerung zur nächsten Zeile Ihres Skripts weitergeht.

Wenn Sie beispielsweise die Geschwindigkeit auf 1 U/min einstellen und `step(2038)` bei einem Motor mit 2038 Schritten aufrufen, würde es eine volle Minute dauern, bis diese Funktion ausgeführt wird. Um eine präzisere Steuerung zu erreichen, wird empfohlen, eine höhere Geschwindigkeit beizubehalten und nur wenige Schritte bei jedem Aufruf von `step()` zu machen.

- **steps:** die Anzahl der Schritte, um den Motor zu drehen – positiv für eine Richtung, negativ für die andere (int).

4.28 Servo

4.28.1 Überblick

In dieser Lektion erkunden Sie die Verwendung von Arduino und Servomotoren. Mit Fokus auf den Arduino Uno und den SG90-Servomotor lernen Sie, wie Sie das Arduino programmieren, um die schwenkende Bewegung des Servos zu steuern. Diese Technik ist in verschiedenen Anwendungen wie Robotik und automatisierten Systemen unerlässlich.

4.28.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

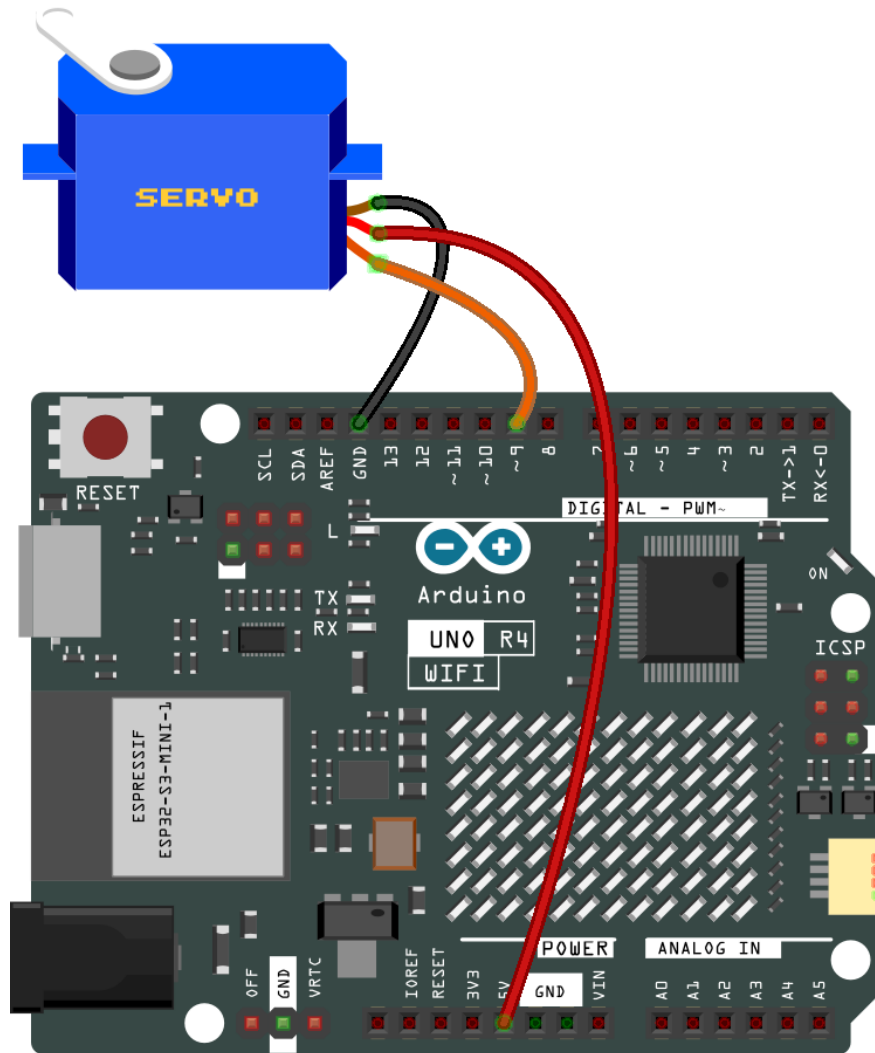
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

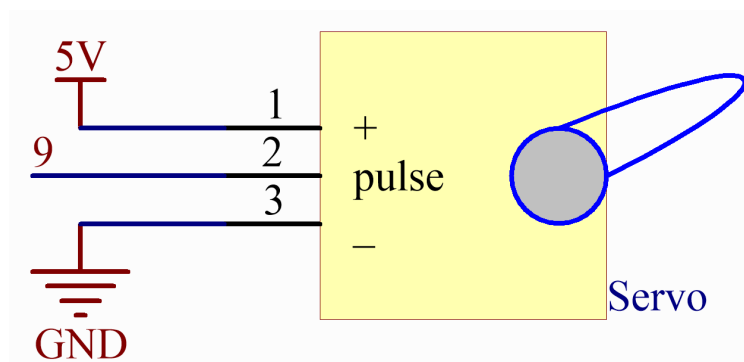
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>Servomotor</i>	

4.28.3 Verdrahtung



4.28.4 Schaltplan



4.28.5 Code

Bemerkung:

- Sie können die Datei 27-servo.ino direkt im Pfad elite-explorer-kit-main\basic_project\27-servo öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

4.28.6 Code-Analyse

1. Hier wird die Servo-Bibliothek eingebunden, die eine einfache Steuerung des Servomotors ermöglicht. Der mit dem Servo verbundene Pin und der Anfangswinkel des Servos werden ebenfalls definiert.

```
#include <Servo.h>
const int servoPin = 9; // Define the servo pin
int angle = 0;          // Initialize the angle variable to 0 degrees
Servo servo;            // Create a servo object
```

2. Die Funktion setup() wird einmal ausgeführt, wenn das Arduino startet. Der Servo wird mit der Funktion attach() an den definierten Pin angehängt.

```
void setup() {
    servo.attach(servoPin);
}
```

3. Die Hauptschleife enthält zwei for-Schleifen. Die erste Schleife erhöht den Winkel von 0 auf 180 Grad, und die zweite Schleife verringert den Winkel von 180 auf 0 Grad. Der Befehl servo.write(angle) setzt den Servo auf den angegebenen Winkel. Die delay(15)-Anweisung bewirkt, dass der Servo 15 Millisekunden wartet, bevor er zum nächsten Winkel übergeht, und steuert so die Geschwindigkeit der Scanbewegung.

```
void loop() {
    // scan from 0 to 180 degrees
    for (angle = 0; angle < 180; angle++) {
        servo.write(angle);
        delay(15);
    }
    // now scan back from 180 to 0 degrees
    for (angle = 180; angle > 0; angle--) {
        servo.write(angle);
        delay(15);
    }
}
```


4.29 Relais

4.29.1 Überblick

Wie wir vielleicht wissen, ist ein Relais ein Gerät, das dazu dient, eine Verbindung zwischen zwei oder mehr Punkten oder Geräten als Reaktion auf das angelegte Eingangssignal herzustellen. Mit anderen Worten, Relais bieten eine Isolation zwischen dem Controller und dem Gerät, da Geräte sowohl mit Wechselstrom als auch mit Gleichstrom arbeiten können. Sie erhalten jedoch Signale von einem Mikrocontroller, der mit Gleichstrom arbeitet, weshalb ein Relais benötigt wird, um die Lücke zu überbrücken. Ein Relais ist äußerst nützlich, wenn Sie eine große Menge an Strom oder Spannung mit einem kleinen elektrischen Signal steuern müssen.

4.29.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

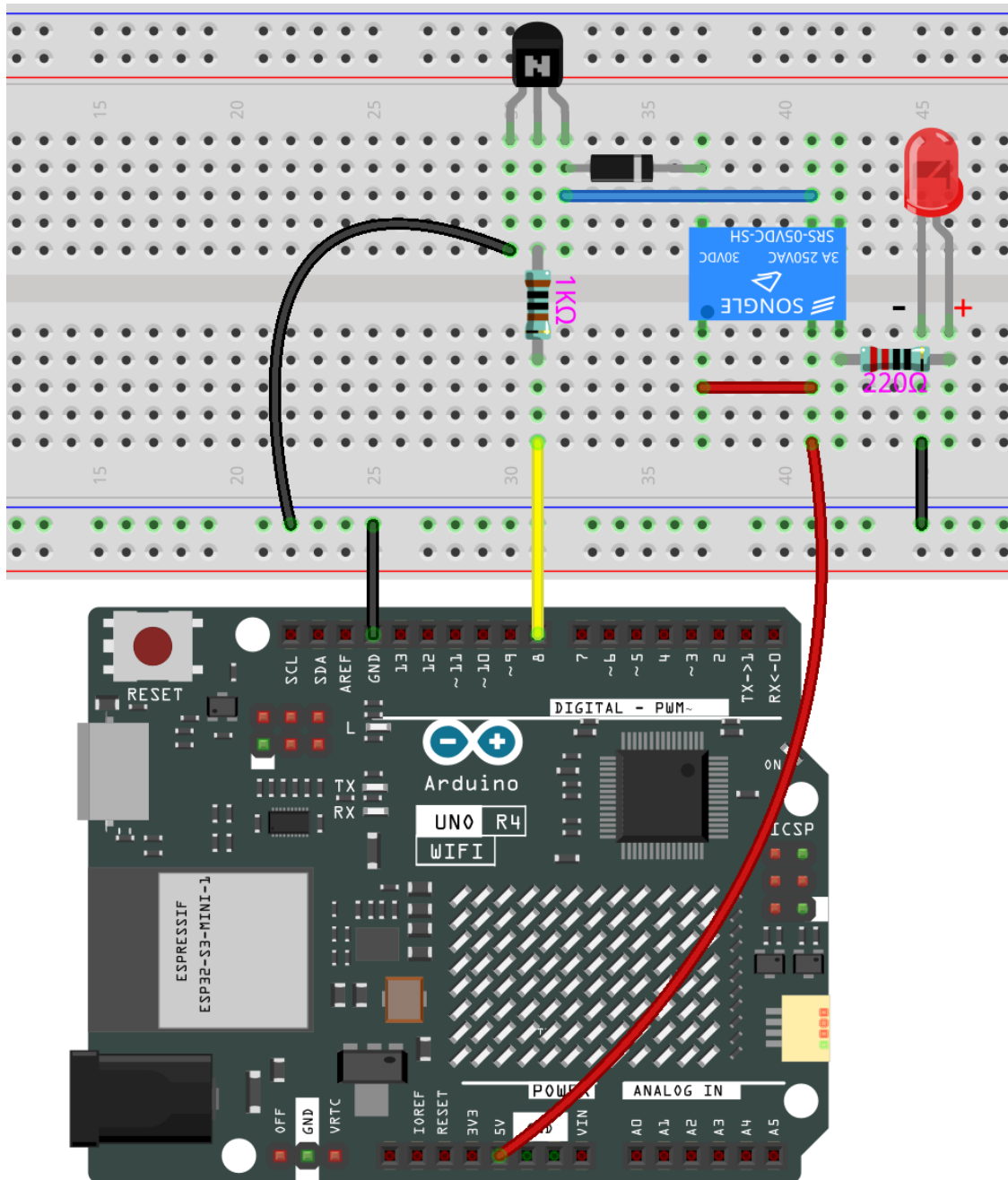
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

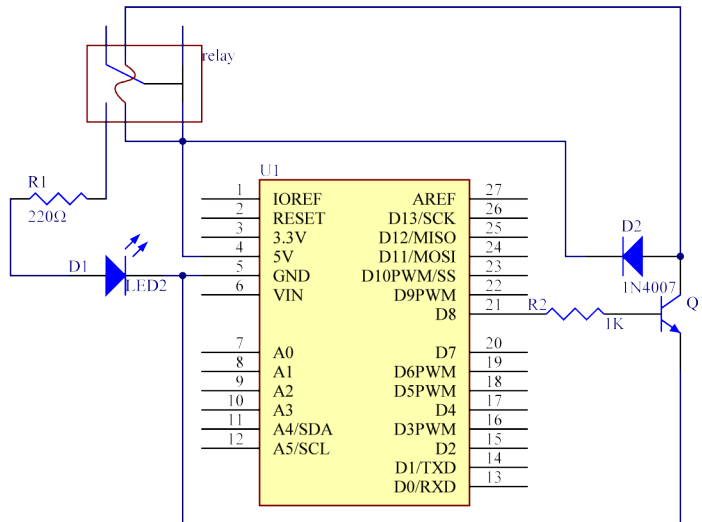
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Relais</i>	
<i>Transistor</i>	
<i>Diode</i>	

4.29.3 Verdrahtung



4.29.4 Schaltplan

Verbinden Sie einen 1K-Widerstand (zur Strombegrenzung, wenn der Transistor aktiviert wird) mit Pin 8 des SunFounder Uno Boards und dann mit einem NPN-Transistor, dessen Kollektor mit der Spule eines Relais verbunden ist und dessen Emittor mit GND; verbinden Sie den normalerweise offenen Kontakt des Relais mit einer LED und dann mit GND. Daher leuchtet die LED auf, wenn Pin 8 ein High-Level-Signal erhält, da der Transistor aktiviert wird und die Spule des Relais leitend macht. Wenn Pin 8 ein Low-Level-Signal erhält, bleibt die LED dunkel.



Funktion der Freilaufdiode: Wenn die Spannungseingabe von High (5V) auf Low (0V) wechselt, ändert sich der Transistor von Sättigung (drei Arbeitsbedingungen: Verstärkung, Sättigung und Abschaltung) zu Abschaltung. Der Strom in der Spule hat plötzlich keinen Durchflussweg mehr. In diesem Moment würde ohne die Freilaufdiode eine Gegen-Elektromotorische Kraft (EMK) an den Enden der Spule erzeugt, mit positivem Pol unten und negativem Pol oben, eine Spannung höher als 100V. Diese Spannung plus die vom Transistor zugeführte Spannung sind groß genug, um ihn zu verbrennen. Daher ist die Freilaufdiode extrem wichtig, um diese Gegen-EMK in Richtung des Pfeils in der Abbildung oben zu entladen, sodass die Spannung des Transistors zu GND nicht höher als +5V (+0.7V) ist.

In diesem Experiment leuchtet die LED auf, wenn das Relais schließt; wenn das Relais öffnet, erlischt die LED.

4.29.5 Code

Bemerkung:

- Sie können die Datei 28-relay.ino direkt im Pfad elite-explorer-kit-main\basic_project\28-relay öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Senden Sie nun ein High-Level-Signal, und das Relais schließt und die LED leuchtet auf; senden Sie ein Low-Level-Signal, und es öffnet sich und die LED erlischt. Zusätzlich können Sie ein Klick-Klack-Geräusch hören, verursacht durch das Öffnen des normalerweise geschlossenen Kontakts und das Schließen des normalerweise offenen Kontakts.

4.29.6 Code-Analyse

```
void loop() {  
  digitalWrite(relayPin, HIGH); // Turn the relay on  
  delay(1000);                 // Wait for one second  
  digitalWrite(relayPin, LOW);  // Turn the relay off  
  delay(1000);                 // Wait for one second  
}
```

Der Code in diesem Experiment ist einfach. Zuerst wird relayPin als High-Level gesetzt und die mit dem Relais verbundene LED leuchtet auf. Dann wird relayPin als Low-Level gesetzt und die LED erlischt.

Chip

4.30 74HC595

4.30.1 Überblick

Im Allgemeinen gibt es zwei Möglichkeiten, eine einzelne 7-Segment-Anzeige anzusteuern. Eine Möglichkeit besteht darin, ihre 8 Pins direkt an acht Ports des Uno-Boards anzuschließen, was wir bereits getan haben. Oder Sie können den 74HC595 an drei Ports des UNO-Boards anschließen und dann die 7-Segment-Anzeige an den 74HC595. In diesem Experiment werden wir Letzteres verwenden. Auf diese Weise können wir fünf Ports sparen – in Anbetracht der begrenzten Ports des Uno-Boards ist das sehr wichtig. Lassen Sie uns jetzt beginnen!

4.30.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

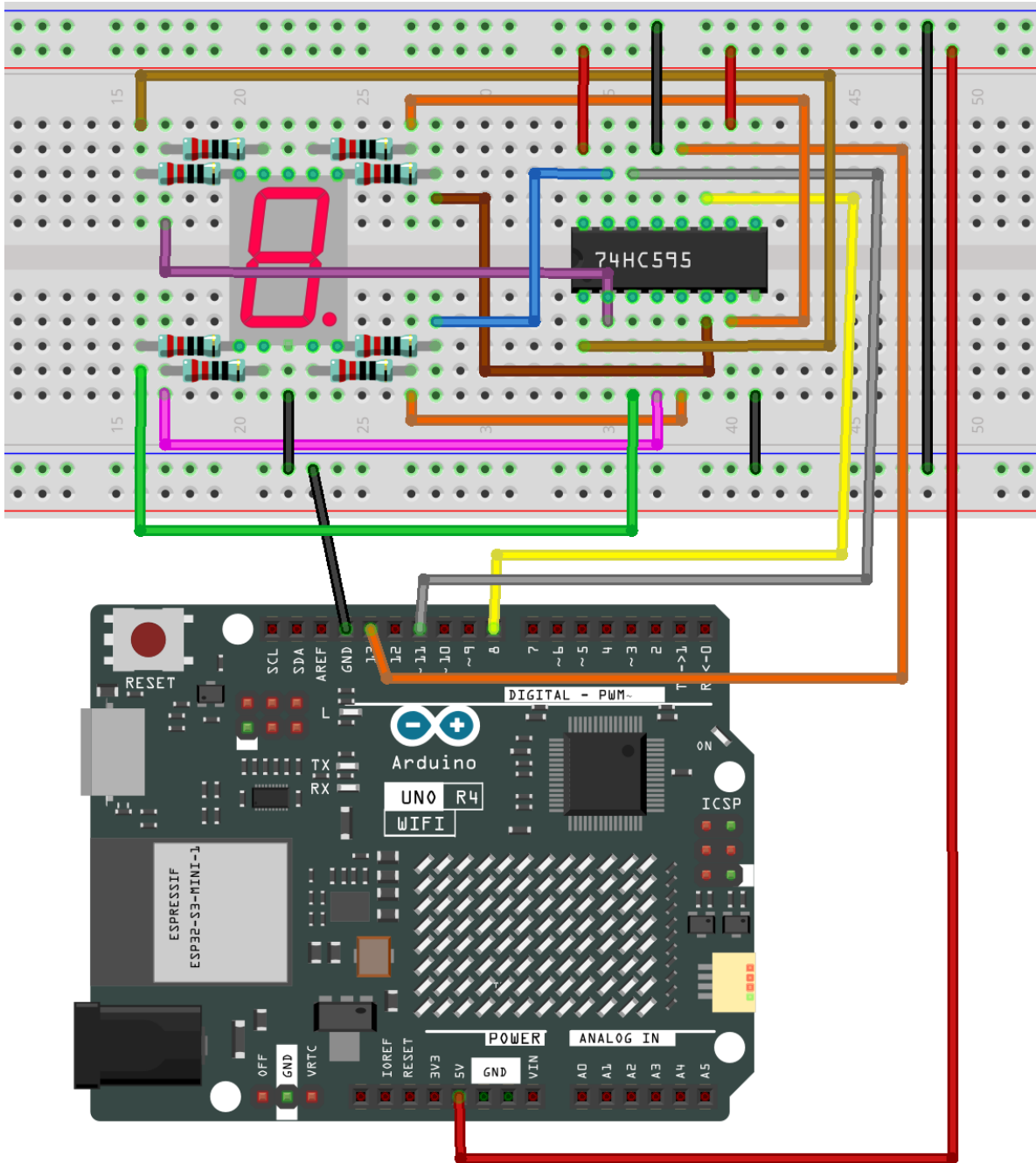
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>7-Segment-Anzeige</i>	
<i>74HC595</i>	

4.30.3 Verdrahtung



4.30.6 Code-Analyse

Setzen der Array-Elemente

```
int dataArray[16] = {252, 96, 218, 242, 102, 182, 190, 224, 254, 246, 238, 62, 156, 122, 158, 142};
```

Dieses Array speichert die Daten der 16 Zeichen von 0 bis F. 252 steht für 0, was Sie selbst berechnen können. Um 0 anzuzeigen, muss das Segment g (das mittlere) der 7-Segment-Anzeige auf niedrigem Niveau (dunkel) sein.

Da das Segment g mit Q1 des 74HC595 verbunden ist, setzen Sie sowohl Q1 als auch DP (den Punkt) auf niedriges Niveau und lassen Sie die restlichen Pins auf hohem Niveau. Daher sind die Werte von Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 1 1 1 1 1 0 0.

Ändern Sie die Binärzahlen in Dezimalzahlen: $1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 252$.

So ist das der Wert für die Anzeige der Zahl 0. Sie können andere Zeichen ähnlich berechnen.

Anzeigen von 0-F in der 7-Segment-Anzeige

```
for(int num = 0; num < 16; num++)
{
    digitalWrite(STcp,LOW); //ground ST_CP and hold low for as long as you are transmitting
    shiftOut(DS,SHcp,MSBFIRST,dataArray[num]);

    //return the latch pin high to signal chip that it
    //no longer needs to listen for information

    digitalWrite(STcp,HIGH); //pull the ST_CPST_CP to save the data

    delay(1000); //wait for a second
}
```

Setzen Sie zuerst STcp auf niedriges und dann auf hohes Niveau. Es erzeugt einen steigenden Flankenimpuls von STcp.

shiftOut() wird verwendet, um ein Byte Daten bitweise auszugeben, was bedeutet, ein Byte Daten in dataArray[num] mit dem DS-Pin in das Schieberegister zu verschieben. MSBFIRST bedeutet, von hohen Bits zu verschieben.

Nachdem digitalWrite(STcp,HIGH) ausgeführt wurde, wird STcp an der steigenden Flanke sein. Zu diesem Zeitpunkt werden die Daten im Schieberegister in das Speicherregister verschoben.

Ein Byte Daten wird nach 8 Mal in das Speicherregister übertragen. Dann werden die Daten des Speicherregisters auf den Bus (Q0-Q7) ausgegeben. Sie sehen ein Zeichen auf der 7-Segment-Anzeige. Dann warten Sie 1000 ms. Danach geht es zurück zur for()-Schleife. Die Schleife wiederholt sich, bis alle Zeichen nacheinander in der 7-Segment-Anzeige nach 16 Mal angezeigt werden.

Erkunden Sie Arduino® UNO R4 WiFi

Das **Arduino Uno R4 WiFi Board** ist die neueste Ergänzung zur Arduino-Familie und bietet eine Reihe fortschrittlicher Funktionen, während es die Robustheit und Vielseitigkeit bewahrt, die Sie erwarten. Dieses Board bringt einen signifikanten Schub in der Verarbeitungsleistung, dank seines 32-Bit Arm® Cortex®-M4 Mikrocontrollers, und es bewahrt die Kompatibilität mit dem Standard-Formfaktor und der Stapelbarkeit der UNO-Serie.

Was ist neu am Uno R4 im Vergleich zum R3?

- **Verbesserte Verarbeitung:** Wechsel von einem 8-Bit-AVR zu einem 32-Bit-Arm® Cortex®-M4-Mikrocontroller.
- **Speicher-Upgrade:** Heben Sie Ihre Projekte mit 32KB SRAM und 256KB NAND-Flash-Speicher an.
- **Erweiterte Konnektivität:** Erleben Sie nahtlose Konnektivität mit USB-C und Wi-Fi.
- **Verbesserte Leistung:** Erreichen Sie mehr mit schnellerer Verarbeitung und größerer Speicherkapazität.
- **Flexible Stromversorgung:** Versorgen Sie Ihr Board mit bis zu 24V für ein breiteres Spektrum an Projekt-Möglichkeiten.

Neben den oben genannten Upgrades führt das R4 WiFi auch die folgenden neuen Funktionen ein:

Neue Funktionen

5.1 Wi-Fi

Das Arduino UNO R4 WiFi ist mit einem eingebauten ESP32-S3-Modul ausgestattet, das Ihnen ermöglicht, sich mit Wi-Fi®-Netzwerken zu verbinden und Netzwerkoperationen durchzuführen. Es unterstützt Protokolle wie HTTPS, MQTT und UDP, die getestet wurden und kompatibel sind.

Im Folgenden werde ich Sie durch zwei Beispiele der Nutzung von Wi-Fi führen:

5.1.1 Mit Wi-Fi verbinden

Dieses Tutorial führt Sie durch die wesentlichen Schritte, um Ihr Arduino-Board mit einem Wi-Fi-Netzwerk zu verbinden. Sie lernen, wie Sie das Wi-Fi-Modul initialisieren, seine Firmware überprüfen und sicher einem Netzwerk mit SSID und Passwort beitreten. Sobald Sie verbunden sind, erfahren Sie, wie Sie wichtige Netzwerkinformationen wie die IP- und MAC-Adressen Ihres Geräts sowie die Signalstärke des Netzwerks direkt über die serielle Konsole überwachen. Dieses Tutorial dient sowohl als praktischer Leitfaden für die Wi-Fi-Konnektivität als auch als Einführung in die Netzwerküberwachung mit Arduino, um eine zuverlässige Wi-Fi-Verbindung herzustellen und aufrechtzuerhalten.

1. Laden Sie den Code hoch

Öffnen Sie die Datei `01-wifi_connect.ino` im Pfad `elite-explorer-kit-main\r4_new_feature\01-wifi_connect` oder kopieren Sie diesen Code in die **Arduino IDE**.

Bemerkung: Wi-Fi® wird über die integrierte `WiFiS3`-Bibliothek aktiviert, die mit dem Arduino UNO R4 Core geliefert wird. Die Installation des Cores installiert automatisch die `WiFiS3`-Bibliothek.

Sie müssen auch `arduino_secrets.h` erstellen oder bearbeiten, ersetzen Sie `SECRET_SSID` und `SECRET_PASS` durch den Namen und das Passwort des Wi-Fi, mit dem Sie sich verbinden möchten. Die Datei sollte enthalten:

```
//arduino_secrets.h header file
#define SECRET_SSID "yournetwork"
#define SECRET_PASS "yourpassword"
```

Öffnen Sie den seriellen Monitor, und Sie werden ähnliche Inhalte wie folgt sehen. Arduino gibt die IP- und MAC-Adressen Ihres Geräts sowie die Signalstärke des Netzwerks aus.



2. Code-Erklärung

1. Einbinden von Bibliotheken und Geheimdaten

```
#include <WiFiS3.h>
#include "arduino_secrets.h"
```

- **WiFiS3** ist eine Bibliothek, die Funktionen für die Wi-Fi-Konnektivität bietet. Die Installation des R4-Cores installiert automatisch die WiFiS3-Bibliothek.
- **arduino_secrets.h** ist eine separate Datei, in der Sie Ihre SSID und Ihr Passwort speichern, damit sie nicht in Ihrem Hauptcode offengelegt werden. Das getrennte Speichern von Netzwerk und Passwort verringert das versehentliche Teilen von Wi-Fi-Anmeldedaten.

2. Globale Variablen deklarieren

```
char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int status = WL_IDLE_STATUS;
```

- **ssid** und **pass** enthalten Ihren Netzwerknamen und Ihr Passwort.
- **status** speichert den aktuellen Status Ihrer Wi-Fi-Verbindung.

3. Funktion setup()

Die serielle Schnittstelle wird mit einer Baudrate von 9600 initialisiert. Die Zeile `while (!Serial);` stellt sicher, dass das Programm wartet, bis die serielle Verbindung hergestellt ist.

```
void setup() {
    //Initialize serial and wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }
    ...
}
```

Anschließend überprüft der Code, ob das Wi-Fi-Modul verfügbar ist oder nicht. Wenn nicht, wird das Programm angehalten und jegliche weitere Ausführung gestoppt.

```
...
// check for the WiFi module:
if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    // don't continue
    while (true);
}
...
```

In diesem Teil des Codes überprüfen wir, ob die Firmware-Version des Uno R4 WiFi auf dem neuesten Stand ist. Wenn nicht, wird eine Aufforderung zum Upgrade angezeigt. Sie können [Aktualisieren der Radio-Modul-Firmware auf Ihrem UNO R4 WiFi-Board](#) für das Firmware-Upgrade konsultieren.

```
...
String fv = WiFi.firmwareVersion();
if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
Serial.println("Please upgrade the firmware");
}
...
```

4. Funktion loop()

```
void loop() {
  // check the network connection once every 10 seconds:
  delay(10000);
  printCurrentNet();
}
```

- Alle 10 Sekunden wird die Funktion `printCurrentNet()` aufgerufen, um die aktuellen Netzwerkinformationen auszudrucken.

Referenz

-

5.1.2 WiFi-gesteuerte LED (Access Point)

Dieses Projekt ermöglicht es Ihnen, eine LED über eine Web-Oberfläche zu steuern. Das Arduino-Board fungiert als WiFi-Zugangspunkt und erstellt sein eigenes lokales Netzwerk, mit dem Sie sich über einen Webbrowser verbinden können. Einmal verbunden, können Sie im Webbrowser die IP-Adresse des Geräts aufrufen, wo Sie Optionen finden, um eine LED (an Pin 13 des Boards angeschlossen) ein- und auszuschalten. Das Projekt bietet Echtzeit-Feedback über den LED-Status über den seriellen Monitor, was das Debuggen und Verstehen des Ablaufs der Operationen erleichtert.

1. Laden Sie den Code hoch

Öffnen Sie die Datei `01-wifi_ap.ino` im Pfad `elite-explorer-kit-main\r4_new_feature\01-wifi_ap` oder kopieren Sie diesen Code in die **Arduino IDE**.

Bemerkung: Wi-Fi® wird über die integrierte WiFiS3-Bibliothek aktiviert, die mit dem Arduino UNO R4 Core geliefert wird. Die Installation des Cores installiert automatisch die WiFiS3-Bibliothek.

Sie müssen auch `arduino_secrets.h` erstellen oder bearbeiten, ersetzen Sie `SECRET_SSID` und `SECRET_PASS` durch den Namen und das Passwort Ihres WiFi-Zugangspunkts. Die Datei sollte enthalten:

```
//arduino_secrets.h header file
#define SECRET_SSID "yournetwork"
#define SECRET_PASS "yourpassword"
```

2. Code-Erklärung

1. Importieren erforderlicher Bibliotheken

Importieren der WiFiS3-Bibliothek für Wi-Fi-Funktionalitäten und `arduino_secrets.h` für sensible Daten wie Passwörter.

```
#include "WiFiS3.h"
#include "arduino_secrets.h"
```

2. Konfiguration und Variableninitialisierung

Definieren Sie WiFi-SSID, Passwort und Schlüsselindex sowie den LED-Pin und den WiFi-Status.

```
char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int keyIndex = 0;
int led = LED_BUILTIN;
int status = WL_IDLE_STATUS;
WiFiServer server(80);
```

3. Funktion setup()

Initialisieren der seriellen Kommunikation und Konfigurieren des Wi-Fi-Moduls.

```
void setup() {
    // ... setup code ...
    // Create access point
    status = WiFi.beginAP(ssid, pass);
    // ... error handling ...
    // start the web server on port 80
    server.begin();
}
```

Wir überprüfen auch, ob die Firmware-Version von Uno R4 WiFi auf dem neuesten Stand ist. Wenn nicht, wird eine Aufforderung zum Upgrade angezeigt. Sie können *Aktualisieren der Radio-Modul-Firmware auf Ihrem UNO R4 WiFi-Board* für das Firmware-Upgrade konsultieren.

```
...
String fv = WiFi.firmwareVersion();
if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial.println("Please upgrade the firmware");
}
...
```

Sie möchten möglicherweise den folgenden Code ändern, um die Standard-IP von Arduino ändern zu können.

```
WiFi.config(IPAddress(192,48,56,2));
```

4. Hauptfunktion loop()

Die Funktion `loop()` im Arduino-Code führt mehrere Schlüsseloperationen aus, insbesondere:

1. Überprüfen, ob ein Gerät sich mit dem Zugangspunkt verbunden oder getrennt hat.
2. Abhören von eingehenden Clients, die HTTP-Anfragen stellen.

3. Lesen der Clientdaten und Ausführen von Aktionen basierend auf diesen Daten – wie das Ein- oder Ausschalten einer LED.

Hier zerlegen wir die Funktion `loop()`, um diese Schritte verständlicher zu machen.

1. Überprüfen des WiFi-Status

Zuerst überprüft der Code, ob sich der WiFi-Status geändert hat. Wenn sich ein Gerät verbunden oder getrennt hat, wird die Information entsprechend auf dem seriellen Monitor angezeigt.

```
if (status != WiFi.status()) {
    status = WiFi.status();
    if (status == WL_AP_CONNECTED) {
        Serial.println("Device connected to AP");
    } else {
        Serial.println("Device disconnected from AP");
    }
}
```

2. Abhören von eingehenden Clients

`WiFiClient client = server.available();` wartet auf eingehende Clients.

```
WiFiClient client = server.available();
```

3. Behandlung von Clientanfragen

Lauscht auf eingehende Clients und liefert ihnen die HTML-Webseite. Wenn ein Benutzer auf den Link „Hier klicken, um die LED einzuschalten“ oder „Hier klicken, um die LED auszuschalten“ auf der bereitgestellten Webseite klickt, wird eine HTTP-GET-Anfrage an den Arduino-Server gesendet. Speziell werden die URLs „`http://yourAddress/H`“ zum Einschalten der LED und „`http://yourAddress/L`“ zum Ausschalten aufgerufen.

```
WiFiClient client = server.available();
if (client) {
    // ...
    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println();
    client.print("<p style='font-size:7vw;'>Click <a href='\"/H\"'>here</a> turn_
↳ the LED on<br></p>");
    client.print("<p style='font-size:7vw;'>Click <a href='\"/L\"'>here</a> turn_
↳ the LED off<br></p>");
    // ...
}
```

Der Arduino-Code lauscht auf diese eingehenden GET-Anfragen. Wenn er am Ende einer eingehenden Textzeile (HTTP-Header) `GET /H` erkennt, setzt er die mit Pin 13 verbundene LED auf HIGH und schaltet sie somit ein. Ähnlich wird die LED auf LOW gesetzt und somit ausgeschaltet, wenn `GET /L` erkannt wird.

```
while (client.connected()) { // loop while the client's connected
    delayMicroseconds(10); // This is required for the Arduino_
↳ Nano RP2040 Connect - otherwise it will loop so fast that SPI will never be_
↳ served.
    if (client.available()) { // if there's bytes to read from the_
↳ client,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

char c = client.read();           // read a byte, then
Serial.write(c);                 // print it out to the serial monitor
if (c == '\n') {                 // if the byte is a newline character
    ...
}
else { // if you got a newline, then clear currentLine:
    currentLine = "";
}
else if (c != '\r') { // if you got anything else but a carriage return_
↪character,
    currentLine += c; // add it to the end of the currentLine
}

// Check to see if the client request was "GET /H" or "GET /L":
if (currentLine.endsWith("GET /H")) {
    digitalWrite(led, HIGH); // GET /H turns the LED on
}
if (currentLine.endsWith("GET /L")) {
    digitalWrite(led, LOW); // GET /L turns the LED off
}
}

```

Referenz

-

5.2 Bluetooth

Ausgestattet mit dem ESP32-Modul bietet das UNO R4 WiFi-Board sowohl Bluetooth® LE als auch Bluetooth® 5-Funktionalitäten und unterstützt Geschwindigkeiten von bis zu 2 Mbps. Das ESP32-Modul kommt mit einer integrierten Trace-Antenne, was die Notwendigkeit einer externen Antenne zur Nutzung der Konnektivitätsfunktionen des Boards überflüssig macht.

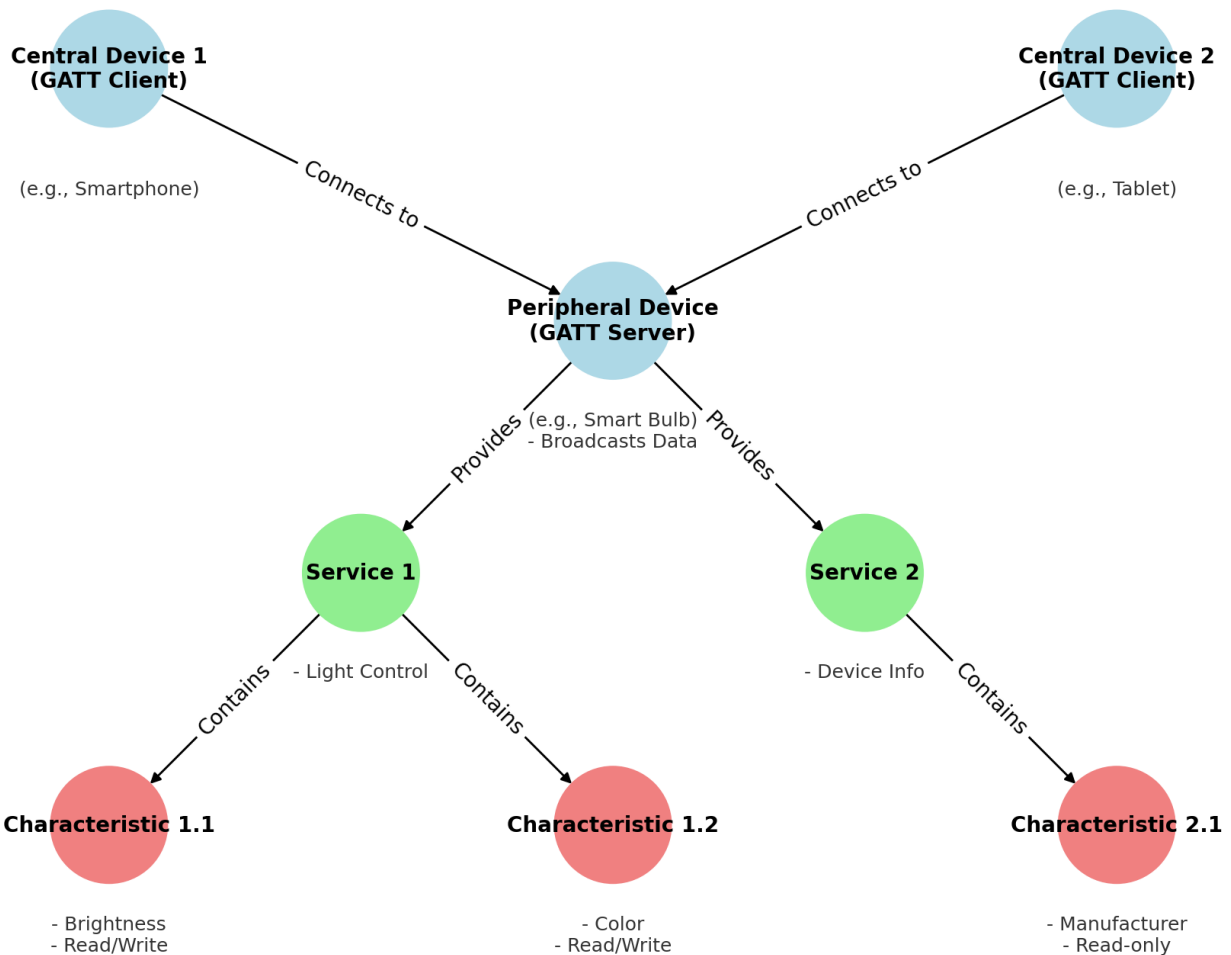
Bemerkung: Die Trace-Antenne im ESP32-Modul wird mit dem Bluetooth®-Modul geteilt, was bedeutet, dass Sie Bluetooth® und Wi-Fi® nicht gleichzeitig verwenden können.

5.2.1 Grundkonzepte von BLE

Bluetooth Low Energy (BLE) ist eine energiesparende drahtlose Kommunikationstechnologie, die speziell für Kurzstreckeninteraktionen entwickelt wurde. Im Unterschied zu klassischem Bluetooth konzentriert sich BLE auf Energieeffizienz und schnelle Verbindung, was es zu einer idealen Wahl für eine Reihe von Anwendungen wie Internet-of-Things (IoT)-Geräte und Gesundheitsüberwachungsausrüstung macht.

BLE-Kommunikationen basieren auf zwei Schlüsselprotokollen: **GATT (Generic Attribute Profile)** und **GAP (Generic Access Profile)**. GATT wird für den Datenaustausch verwendet, während GAP für die Geräteerkennung und -verbindung verantwortlich ist.

BLE (Bluetooth Low Energy)



Peripheriegeräte (typischerweise GATT-Server)

Im BLE-Netzwerk übertragen **peripheral devices** hauptsächlich Daten, um von zentralen Geräten (typischerweise als GATT-Clients agierend) entdeckt und darauf zugegriffen zu werden. Solche Geräte sind in der Regel Sensoren oder kleine Hardware wie Herzfrequenzmesser, Temperatursensoren oder intelligente Glühbirnen.

Im BLE-Kommunikationsmodell bieten Peripheriegeräte häufig einen oder mehrere **services** an, die jeweils eine Reihe von **Characteristics** enthalten. Diese Dienste und Charakteristiken ermöglichen gemeinsam spezifische Funktionalitäten oder Anwendungsfälle, die es zentralen Geräten erlauben, relevante Daten zu lesen oder zu manipulieren.

- **Services**

In BLE fungieren Dienste als hochrangige Abstraktionen, die verwendet werden, um verwandte Charakteristiken zu organisieren und einzukapseln. Dienste in BLE können in Standarddienste und benutzerdefinierte Dienste eingeteilt werden, abhängig von ihrer Herkunft und ihrem Zweck.

- Standarddienste: Definiert von der Bluetooth SIG (Bluetooth Special Interest Group), sind diese für spezifische Funktionen gedacht. Zum Beispiel der Herzfrequenzdienst für Herzfrequenzmesser, der Geräteinformationsdienst, der Hersteller-, Modell- und Versionsdetails bereitstellt, und der Batteriedienst, der den Batteriestand und -status anzeigt.
- Benutzerdefinierte Dienste: Diese werden von Entwicklern oder Geräteherstellern definiert, um die An-

forderungen spezifischer Anwendungen oder Geräte zu erfüllen. Beispielsweise könnte ein Hersteller von Smart-Home-Geräten einen benutzerdefinierten Dienst definieren, um die Farbe und Helligkeit des Lichts zu steuern.

- **Characteristics**

Charakteristiken in BLE sind die grundlegenden Dateneinheiten, die von den Peripheriegeräten bereitgestellt werden. Sie sind innerhalb eines Dienstes enthalten und definieren verschiedene Datentypen und die Operationen, die an ihnen durchgeführt werden können. Jede Charakteristik wird durch eine UUID identifiziert und hat eine Reihe von zugehörigen Attributen wie Wert, Beschreibung und Berechtigungen.

- **Berechtigungen:** In BLE ist jede Charakteristik mit einem Satz von Berechtigungen verbunden, die bestimmen, ob die Charakteristik lesbar, beschreibbar oder benachrichtigungsfähig ist. Dies hilft, die Daten zu sichern und zu definieren, wie mit ihnen interagiert wird.

- **UUID**

Dienste, Charakteristiken und Beschreiber werden kollektiv als Attribute identifiziert, wobei jedes ein einzigartiges UUID besitzt. Die Bluetooth SIG hat eine Reihe von UUIDs für Standardattribute reserviert. Diese UUIDs werden im BLE-Protokoll normalerweise als 16-Bit- oder 32-Bit-Identifikatoren dargestellt, um Effizienz zu gewährleisten, anstatt der 128 Bit, die für eine vollständige UUID erforderlich wären. Zum Beispiel wird der Dienst für Geräteinformationen durch den Kurzcode 0x180A repräsentiert.

Central Devices (Typically GATT Clients)

Central devices im BLE-Netzwerk suchen nach nahegelegenen Peripheriegeräten und stellen Verbindungen her, um Daten zu erwerben oder zu steuern. Diese Geräte sind in der Regel komplexer und funktionsreicher, wie Smartphones, Tablets oder spezialisierte Gateway-Hardware. Sie sind verantwortlich für das Auffinden von Peripheriegeräten, deren Verbindung und den Zugriff oder das Abonnieren von Diensten und Charakteristiken, die von den Peripheriegeräten angeboten werden, um verschiedene Anwendungen zu bedienen oder spezifische Probleme zu lösen.

Zentrale Geräte interagieren auf folgende Weise mit Charakteristiken:

- **Lesen:** Fordert das Peripheriegerät auf, den aktuellen Wert einer Charakteristik zu senden. Dies wird häufig für Charakteristiken verwendet, die sich nicht oft ändern, wie Konfigurationseinstellungen oder Versionsnummern.
- **Schreiben:** Ändern Sie den Wert einer Charakteristik, typischerweise verwendet für befehlsähnliche Operationen, wie das Anweisen eines Peripheriegeräts, einen Motor ein- oder auszuschalten.
- **Abonnieren:** Fordert das Peripheriegerät auf, kontinuierlich aktualisierte Werte einer Charakteristik zu senden und eliminiert die Notwendigkeit für das zentrale Gerät, diese Daten wiederholt anzufordern.

5.2.2 Beispiel: Bluetooth-gesteuerte LED

In diesem Beispiel fungiert das Arduino als Peripheriegerät in einem Bluetooth Low Energy (BLE)-Netzwerk. Es bietet einen benutzerdefinierten BLE-Dienst an, der zur Steuerung einer Onboard-LED entwickelt wurde. Dieser Dienst beinhaltet eine Charakteristik, die von einem zentralen Gerät, wie einem Smartphone, gelesen und beschrieben werden kann. Sobald das zentrale Gerät eine Verbindung zum Arduino herstellt, kann es den LED-Status durch Schreiben dieser Charakteristik ändern. Der serielle Monitor des Arduino zeigt Debugging-Informationen an, einschließlich des aktuellen Zustands der LED und der MAC-Adresse des verbundenen zentralen Geräts.

Code hochladen

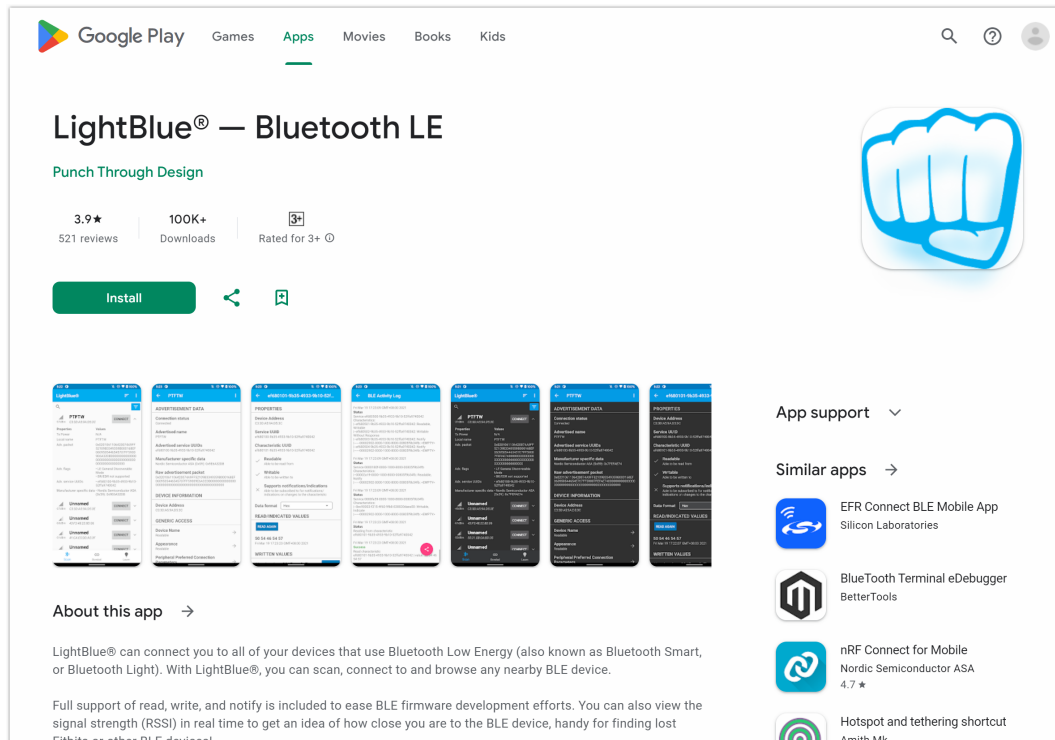
Öffnen Sie die Datei „02-bluetooth.ino“, die sich unter „elite-explorer-kit-mainr4_new_feature02-bluetooth“ befindet, oder fügen Sie den folgenden Code in Ihre Arduino IDE ein.

Verbindung des Arduino R4 über Bluetooth

Um mit den in diesem Sketch erstellten Diensten und Charakteristiken zu interagieren, sollten wir eine generische Bluetooth® Low Energy Central App wie LightBlue (verfügbar für iOS und Android) oder nRF Connect (für Android) nutzen.

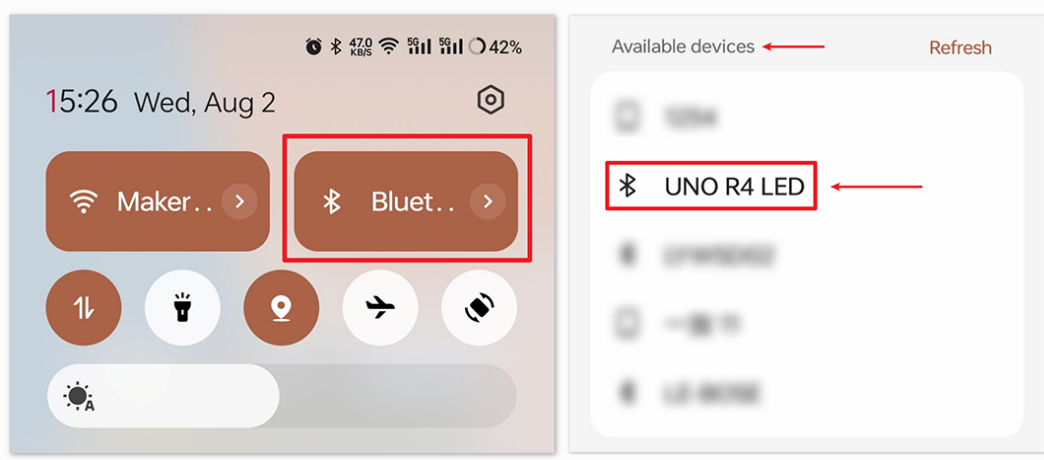
Wir nehmen LightBlue als Beispiel, um zu demonstrieren, wie man die LED des Arduino über Bluetooth steuert.

1. Laden Sie die **LightBlue**-App aus dem (für iOS) oder (für Android) herunter.



2. Verbindung des Arduino mit Ihrem Smartphone über Bluetooth

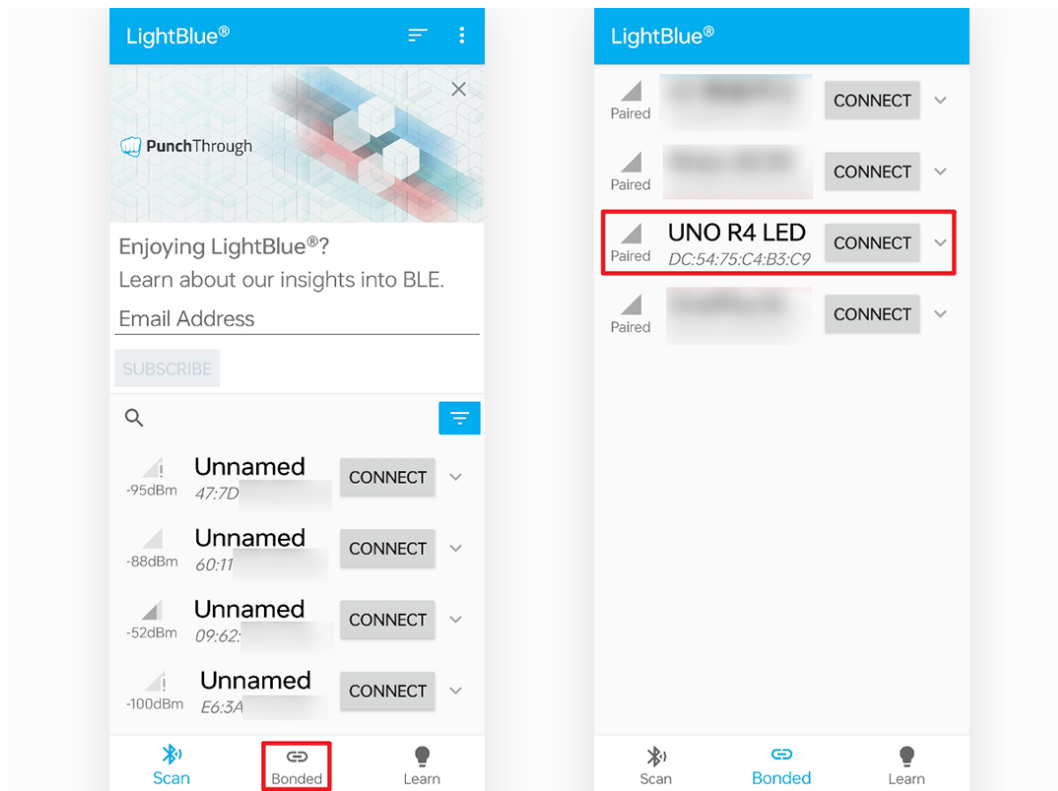
Gehen Sie zu Ihren Bluetooth-Einstellungen und suchen Sie das Gerät namens „UNO R4 LED“. Verbinden Sie sich damit.



3. Interaktion mit dem Arduino über Bluetooth mit LightBlue

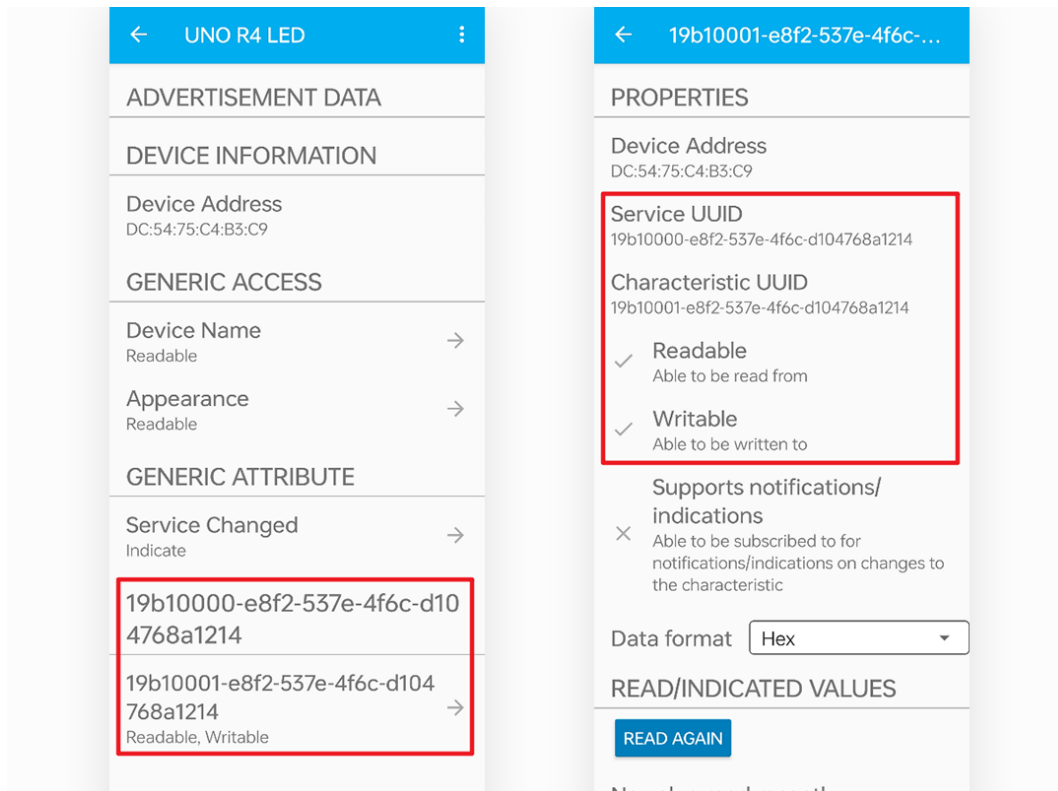
Starten Sie LightBlue und tippen Sie auf den Tab **Bonded** unten in der Schnittstelle. Hier sehen Sie eine Liste der BLE-Geräte, mit denen Ihr Smartphone zuvor gekoppelt wurde. Suchen Sie nach **UNO R4 LED** und tippen

Sie auf **CONNECT**.

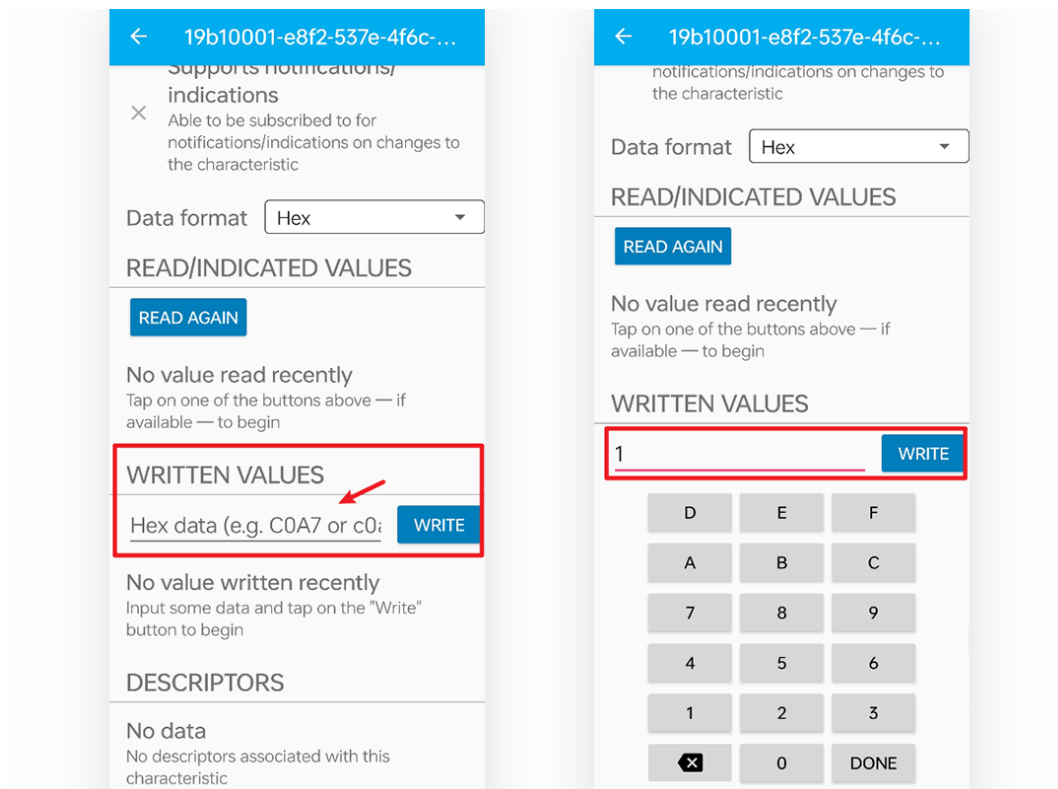


Einmal verbunden, erhalten Sie Zugriff auf detaillierte Informationen über das Bluetooth-Gerät „UNO R4 LED“. Scrollen Sie nach unten, um „ledService (19B10000-E8F2-537E-4F6C-D104768A1214)“ und „switchCharacteristic (19B10001-E8F2-537E-4F6C-D104768A1214)“ zu finden.

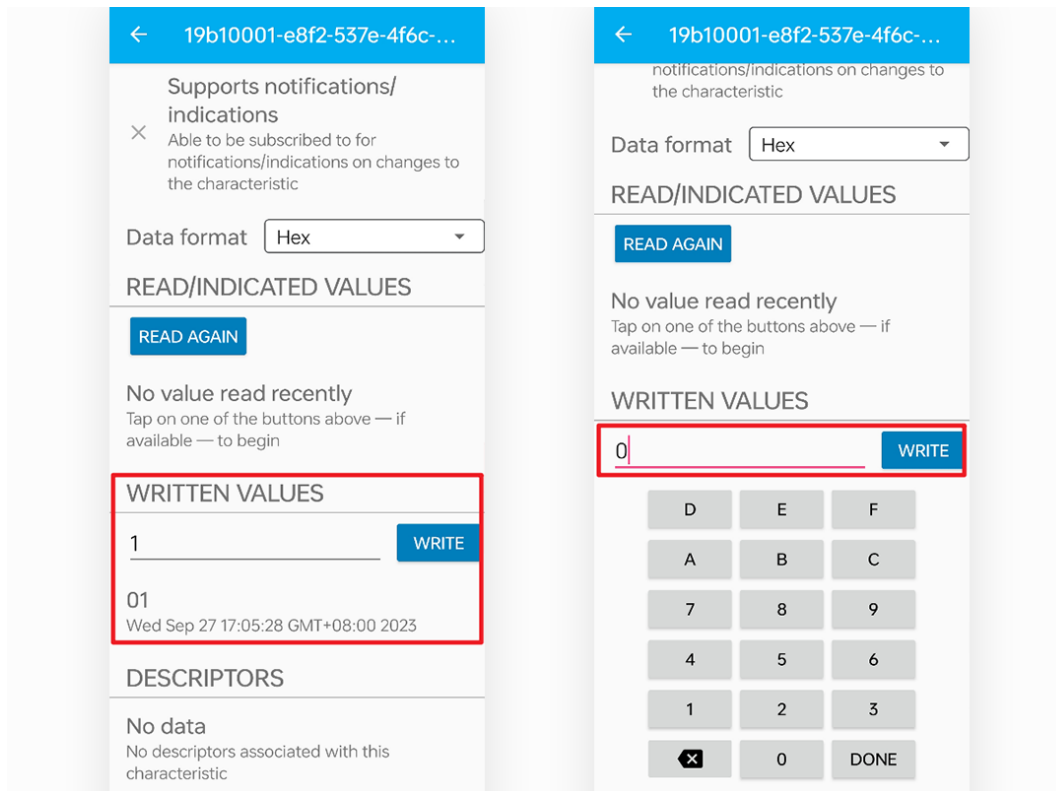
Tippen Sie auf die Charakteristik 19B10001-E8F2-537E-4F6C-D104768A1214. Sie werden feststellen, dass diese Charakteristik sowohl lesbar als auch beschreibbar ist, sodass Sie sowohl von ihr lesen als auch auf sie schreiben können.



Scrollen Sie weiter zum Abschnitt **WRITTEN VALUES**. Geben Sie „1“ in das Textfeld ein, um den Charakteristikwert auf 1 zu setzen, was die **eingebaute LED des Arduino R4 einschaltet**.



Ebenso können Sie diesen Wert auf „0“ setzen, um die **eingebaute LED auszuschalten**.



Code-Erklärung

1. BLE und LED initialisieren

Bemerkung: Beim Definieren von Diensten und Charakteristiken müssen wir UUIDs verwenden, um sie zu identifizieren. Um UUID-Konflikte zu vermeiden und die Nutzung für Sie zu erleichtern, können Sie das UUID-Generierungstool unter verwenden.

```
#include <ArduinoBLE.h>
BLEService ledService("19B10000-E8F2-537E-4F6C-D104768A1214"); // Bluetooth® Low Energy LED Service
BLEByteCharacteristic switchCharacteristic("19B10001-E8F2-537E-4F6C-D104768A1214", BLERead | BLEWrite);
const int ledPin = LED_BUILTIN; // pin to use for the LED
```

- Die ArduinoBLE-Bibliothek einbinden.
- Den BLE-Dienst und die Charakteristik definieren.
- Den eingebauten LED-Pin mit ledPin zuweisen.

2. Funktion setup()

```
void setup() {
  Serial.begin(9600);
  while (!Serial);
  pinMode(ledPin, OUTPUT);
  if (!BLE.begin()) {
    Serial.println("starting Bluetooth® Low Energy module failed!");
  }
}
```

(Fortsetzung auf der nächsten Seite)

```
    while (1);
}
BLE.setLocalName("UNO R4 LED");
BLE.setAdvertisedService(ledService);
ledService.addCharacteristic(switchCharacteristic);
BLE.addService(ledService);
switchCharacteristic.writeValue(0);
BLE.advertise();
Serial.println("BLE LED Peripheral");
}
```

- Serielle Kommunikation initialisieren.
- Den LED-Pin als Ausgang setzen.
- BLE initialisieren und den Dienst sowie die Charakteristiken hinzufügen.
- BLE-Werbung starten.

3. loop() Function

```
void loop() {
  BLEDevice central = BLE.central();
  if (central) {
    Serial.print("Connected to central: ");
    Serial.println(central.address());
    while (central.connected()) {
      if (switchCharacteristic.written()) {
        if (switchCharacteristic.value()) {
          Serial.println("LED on");
          digitalWrite(ledPin, HIGH);
        } else {
          Serial.println("LED off");
          digitalWrite(ledPin, LOW);
        }
      }
    }
  }
  Serial.print("Disconnected from central: ");
  Serial.println(central.address());
}
```

- Auf Verbindungen von BLE-Zentralgeräten lauschen.
- Wenn ein Zentralgerät verbunden ist, den Charakteristikwert lesen, um die LED zu steuern. Wenn ein Wert ungleich 0 empfangen wird, die LED einschalten. Wenn 0 empfangen wird, die LED ausschalten.

Referenz

-

5.3 Echtzeituhr

Die RTC (Real-Time Clock, Echtzeituhr) ist in den Mikrocontroller des UNO R4 WiFi (RA4M1) integriert. Die RTC ist ein autonomes Uhrenmodul, das auch dann funktioniert, wenn die Hauptstromversorgung unterbrochen ist, dank einer Backup-Stromquelle wie einer Batterie. Dies macht die RTC äußerst vielseitig einsetzbar, beispielsweise für die Planung zeitgesteuerter Aufgaben in Hausautomatisierungssystemen oder zur Zeitstempelung einzelner Datenpunkte in Datenlogger-Anwendungen.

Bemerkung: Der UNO R4 WiFi verfügt über einen VRTC-Pin, der den Betrieb der integrierten RTC auch bei Stromausfall des Boards aufrechterhält. Um diese Funktion zu nutzen, sollte eine Spannung zwischen 1,6 und 3,6 V an den VRTC-Pin angelegt werden.

5.3.1 Geplante Wiederkehrende Aufgaben

In bestimmten Anwendungsfällen müssen Sie möglicherweise spezifische Aufgaben in regelmäßigen Abständen ausführen. Um periodische Unterbrechungen einzurichten, müssen Sie zunächst eine periodische Rückruffunktion initialisieren. Unten finden Sie ein Arduino-Codebeispiel, das eine periodische Unterbrechung verwendet, um eine LED alle 2 Sekunden blinken zu lassen.

Code Hochladen

Öffnen Sie die Datei `03-rtc.ino`, die sich im Verzeichnis `elite-explorer-kit-main\r4_new_feature\03-rtc` befindet, oder fügen Sie den folgenden Code in Ihre Arduino IDE ein.

Code-Erklärung

1. Initialisierung von Komponenten und Bibliotheken

```
#include "RTC.h"
volatile bool irqFlag = false;
bool ledState = false;
const int led = LED_BUILTIN;

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
  RTC.begin();
}
```

- `#include "RTC.h"`: Bindet die RTC-Bibliothek ein.
- `volatile bool irqFlag = false;`: Deklariert eine volatile Boolesche Flagge. `volatile` stellt sicher, dass die Variable durch einen Interrupt sicher zugegriffen oder geändert werden kann. Weitere Details zu `volatile` finden Sie in der [Arduino-Volatile-Dokumentation](#).
- `bool ledState = false;`: Deklariert eine Boolesche Variable, um den Zustand der LED zu verfolgen.
- `const int led = LED_BUILTIN;`: Deklariert eine Konstante für den eingebauten LED-Pin.
- `pinMode(led, OUTPUT);`: Stellt den LED-Pin als Ausgang ein.
- `Serial.begin(9600);`: Initialisiert die serielle Kommunikation mit 9600 Baud.
- `RTC.begin();`: Initialisiert die RTC.

2. Einrichtung der Echtzeituhr

```
RTCTime mytime(30, Month::JUNE, 2023, 13, 37, 00, DayOfWeek::WEDNESDAY,
↪SaveLight::SAVING_TIME_ACTIVE);
RTC.setTime(mytime);
```

- `RTCTime mytime(...)`;: Erstellt ein `RTCTime`-Objekt und initialisiert es mit einem spezifischen Datum und Uhrzeit.
- `RTC.setTime(mytime)`;: Stellt die RTC mit der initialisierten Zeit ein.

Periodische Rückruffunktion Einrichten und Überprüfen

```
if (!RTC.setPeriodicCallback(periodicCallback, Period::ONCE EVERY 2_SEC)) {
  Serial.println("ERROR: periodic callback not set");
}

void loop() {
  if (irqFlag) {
    Serial.println("Timed CallBack");
    ledState = !ledState;
    digitalWrite(led, ledState);
    irqFlag = false;
  }
}

void periodicCallback() {
  irqFlag = true;
}
```

- `RTC.setPeriodicCallback(...)`;: Legt eine periodische Rückruffunktion fest, die alle 2 Sekunden ausgelöst wird. Der Zeitraum kann mit den folgenden Enumerationen angegeben werden:
 - `ONCE EVERY 2_SEC`
 - `ONCE EVERY 1_SEC`
 - `N2_TIMES EVERY_SEC`
 - `N4_TIMES EVERY_SEC`
 - `N8_TIMES EVERY_SEC`
 - `N16_TIMES EVERY_SEC`
 - `N32_TIMES EVERY_SEC`
 - `N64_TIMES EVERY_SEC`
 - `N128_TIMES EVERY_SEC`
 - `N256_TIMES EVERY_SEC`
- `void loop() {...}`: Überprüft, ob der Rückruf ausgelöst wurde. Wenn ja, ändert den Zustand der LED.
- `void periodicCallback() {...}`: Die Rückruffunktion setzt `irqFlag = true`, wenn sie ausgelöst wird.

Referenz

-

5.4 12x8 LED-Matrix

Das Arduino UNO R4 WiFi ist mit einer integrierten 12x8 LED-Matrix ausgestattet, die so programmiert werden kann, dass sie eine Vielzahl von Grafiken, Animationen darstellt, als Schnittstelle dient oder sogar Spielerlebnisse ermöglicht.

In diesem Leitfaden bieten wir ein einfaches Beispiel, um Ihnen zu helfen, Ihr gewünschtes Muster auf der LED-Matrix anzuzeigen.

5.4.1 Speicherung von LED-Matrix-Daten in Arduino

Um die LED-Matrix zu nutzen, benötigen Sie die Bibliothek `Arduino_LED_Matrix`, die zusammen mit dem Renesas-Kern installiert wird.

Die LED-Matrix-Bibliothek für das UNO R4 WiFi funktioniert, indem sie Frames in einen Puffer erstellt und lädt, um sie anzuzeigen. Ein Frame, auch bekannt als „Bild“, stellt dar, was aktuell auf der Matrix angezeigt wird. In einer Animation, die aus mehreren Bildern besteht, wird jedes Bild als Frame betrachtet.

Um die 12x8 LED-Matrix auf dem UNO R4 WiFi zu steuern, werden mindestens 96 Bit Speicherplatz benötigt. Die Bibliothek bietet dafür zwei Ansätze.

Ein Ansatz verwendet ein zweidimensionales Array, mit Nullen und Einsen, um darzustellen, ob die entsprechende LED aus oder an ist. Jede Zahl entspricht einer LED auf der LED-Matrix. Das folgende Array veranschaulicht ein herzförmiges Muster.

```
// Use a two-dimensional array to represent a 12x8 LED matrix.
byte frame[8][12] = {
  { 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0 },
  { 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0 },
  { 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0 },
  { 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0 },
  { 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
};
```

Ein anderer Ansatz verwendet ein Array aus 32-Bit-Ganzzahlen, um den Status der LED-Matrix zu verwalten. Diese Methode ist kompakter, aber etwas komplexer. Jedes `unsigned long` speichert 32 Bit. Daher benötigen Sie für eine 12x8 LED-Matrix, die 96 LEDs enthält, mindestens drei `unsigned long` Variablen.

1. Jedes `unsigned long` enthält 32 Bits, und Sie können diese Bits als den Zustand eines bestimmten Teils in einer LED-Matrix betrachten.
2. Diese `unsigned long` Variablen bilden ein Array, das den gesamten Zustand der LED-Matrix umfasst.

Hier ist ein Code-Snippet, das drei `unsigned long` Variablen verwendet, um eine 12x8 LED-Matrix darzustellen.

```
// Use an array of 32-bit integers to store the LED matrix.
unsigned long frame[] = {
  0x3184a444, // State of the first 32 LEDs
  0x42081100, // State of the next 32 LEDs
  0xa0040000 // State of the last 32 LEDs
};
```

Um die LED-Zustände besser zu visualisieren, können diese Werte in eine Binärform umgewandelt werden, wobei jedes Bit sequenziell den Zustand jeder LED von links nach rechts und von oben nach unten darstellt. Eine 0 zeigt an, dass sie aus ist, und eine 1, dass sie an ist.

```
0x3184a444 -> 110001100001001010010001000100
0x42081100 -> 10000100000100000001000100000000
0xa0040000 -> 10100000000001000000000000000000
```

5.4.2 Anzeigemuster auf der LED-Matrix

Sobald Ihr Muster fertig ist, besteht der nächste Schritt darin, diese Daten auf die 12x8 LED-Matrix zu übertragen. Dies beinhaltet normalerweise das Aufrufen von Bibliotheksfunktionen und das Übergeben des Arrays oder der Variablen, die die LED-Zustände enthalten, an diese Funktionen.

1. Verwendung eines zweidimensionalen Arrays

Um das Muster, das in einem 2D-Array gespeichert ist, anzuzeigen, können Sie den folgenden Code verwenden:

```
#include <Arduino_LED_Matrix.h>

ArduinoLEDMatrix matrix;

// Pre-defined 2D array
byte frame[8][12] = {
  { 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0 },
  { 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0 },
  { 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0 },
  { 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0 },
  { 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
};

void setup() {
  // Initialize LED matrix
  matrix.begin();
}

void loop() {
  // Display pattern on the LED matrix
  matrix.renderBitmap(frame, 8, 12);
  delay(1000);
}
```

In diesem Code verwenden wir die Funktion `matrix.renderBitmap(frame, 8, 12);`, um die LED-Matrix anzuzeigen. Hierbei stehen 8 und 12 jeweils für die Zeilen und Spalten der LED-Matrix.

2. Verwendung eines Arrays aus 32-Bit-Ganzzahlen

Um das in einem Array von `unsigned long` gespeicherte Muster anzuzeigen, verwenden Sie den folgenden Code:

```
#include "Arduino_LED_Matrix.h"

ArduinoLEDMatrix matrix;

void setup() {
  matrix.begin();
}

const uint32_t heart[] = {
  0x3184a444,
  0x44042081,
  0x100a0040
};

void loop(){
  matrix.loadFrame(heart);
  delay(500);
}
```

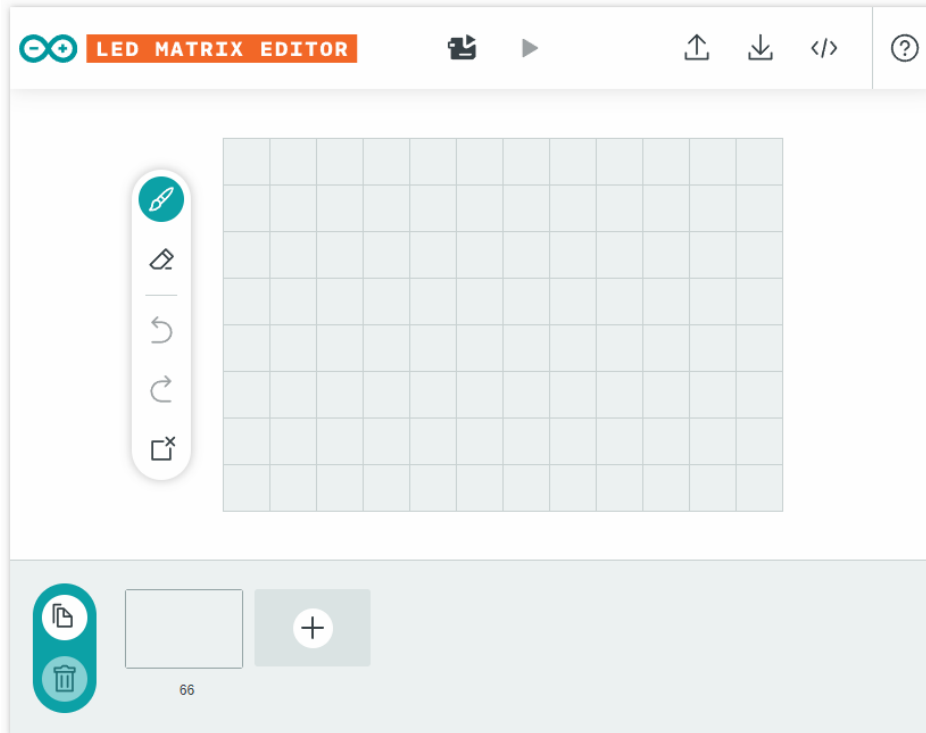
In diesem Fall müssen wir die Funktion `matrix.loadFrame(heart)` verwenden, um das Muster auf der LED-Matrix anzuzeigen.

5.4.3 Arduino LED-Matrix-Editor

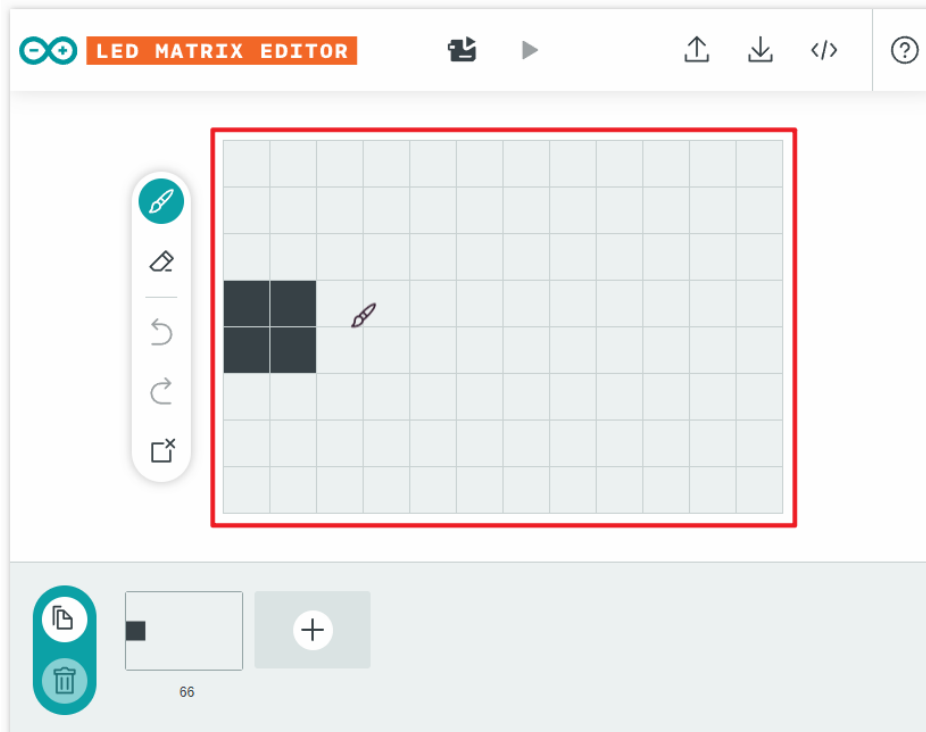
Ich empfehle, ein Array aus `unsigned long` zu verwenden, um den Zustand der LED-Matrix zu speichern, da es Speicherplatz auf dem Arduino spart. Obwohl diese Methode vielleicht nicht sehr intuitiv ist, können Sie den als Hilfsmittel verwenden, der Ihnen hilft, ein Array aus `unsigned long` zu generieren.

Mit dem und der Bibliothek `Arduino_LED_Matrix` können Sie bequem Symbole oder Animationen erstellen und auf dem UNO R4 WiFi-Board anzeigen. Alles, was Sie tun müssen, ist zu zeichnen, die `.h`-Datei herunterzuladen und die Funktion `matrix.play()` in Ihrem Sketch aufzurufen, um Ihr nächstes Projekt einfach zu realisieren.

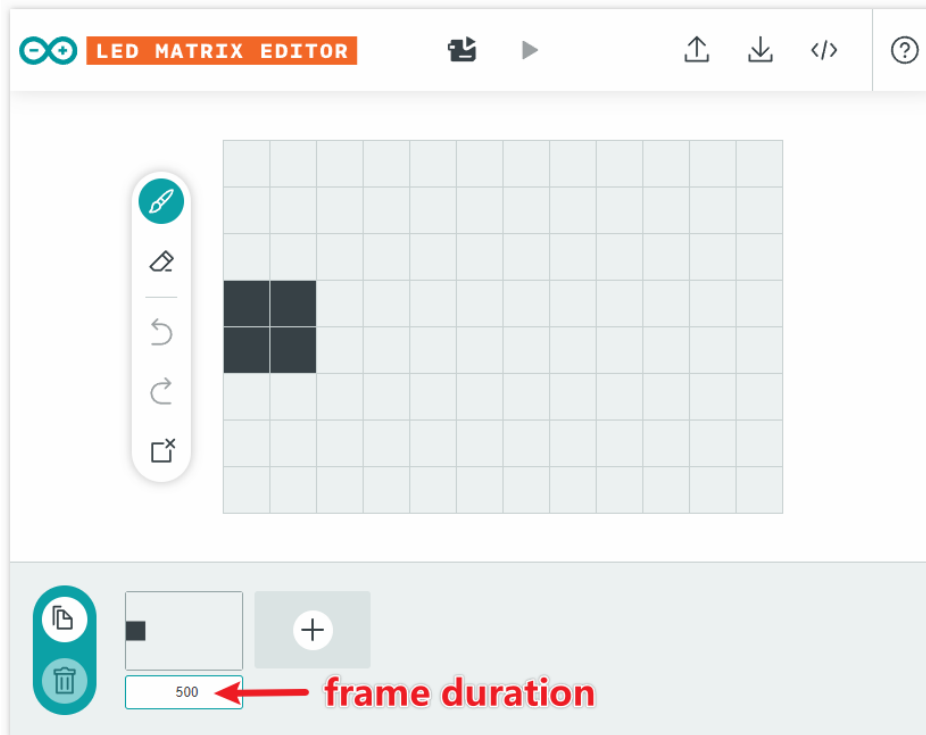
1. Öffnen Sie den LED-Matrix-Editor



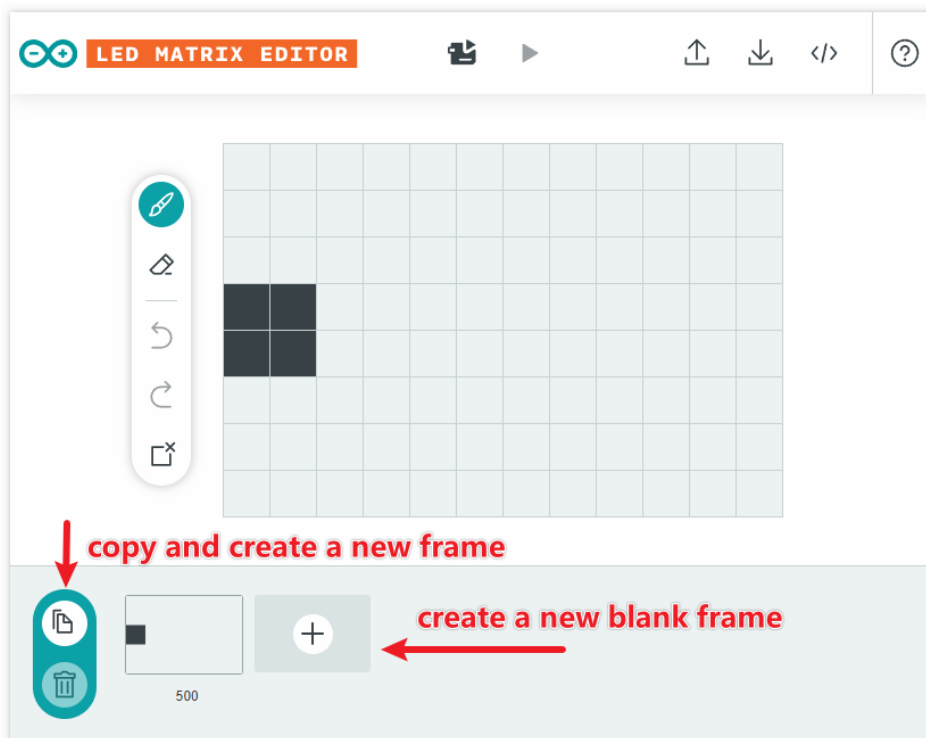
2. Zeichnen Sie Ihr Muster auf die zentrale Leinwand



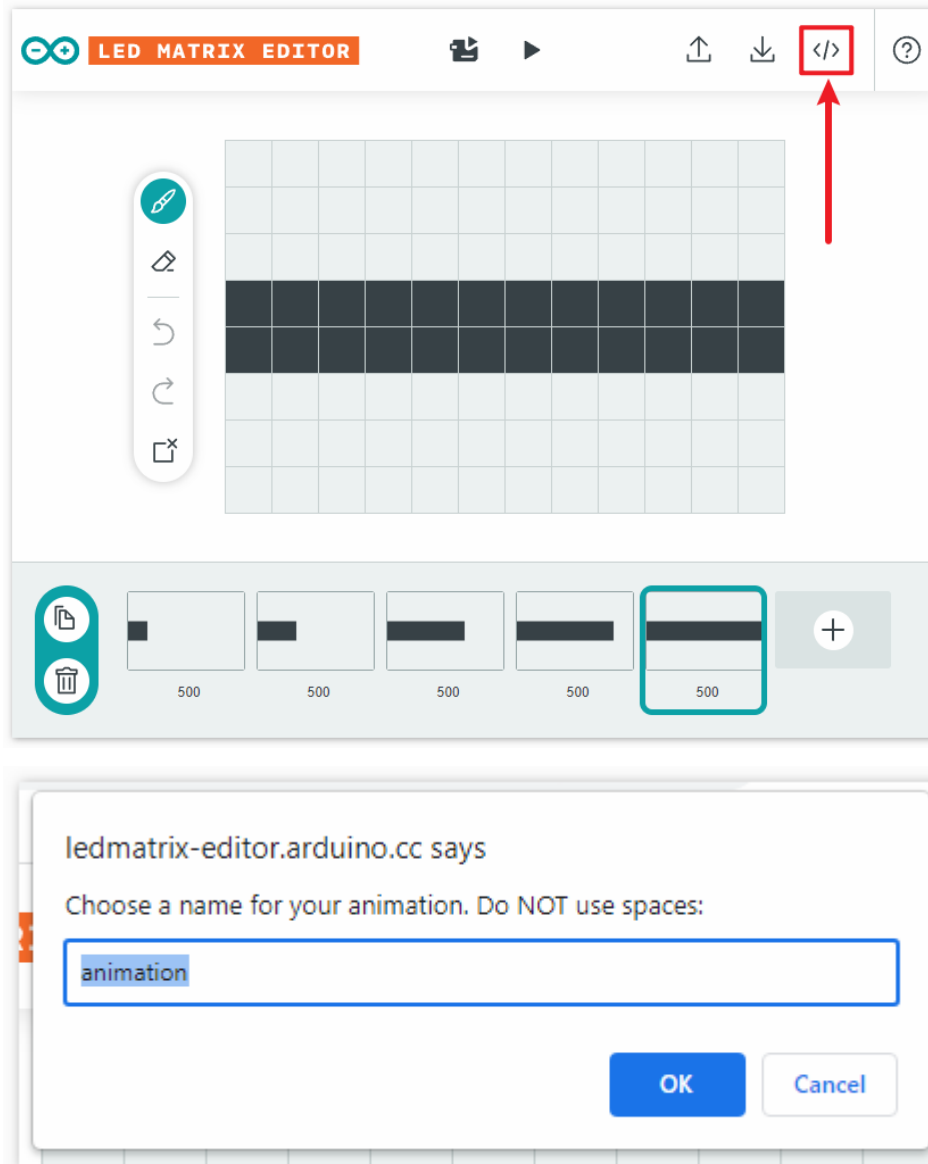
3. Stellen Sie das Frame-Intervall in Millisekunden ein



4. Sie können einen neuen leeren Frame erstellen oder einen neuen Frame aus dem aktuellen Frame kopieren und erstellen.



5. Exportieren Sie die .h-Headerdatei

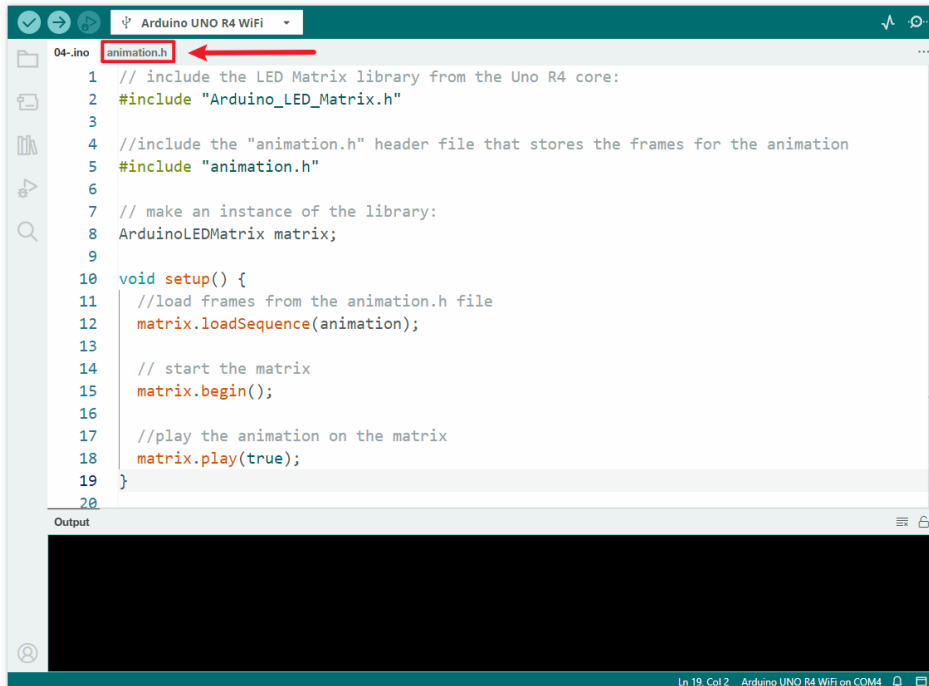


Nachdem Sie auf OK geklickt haben, erhalten Sie eine Datei mit dem Namen `animation.h`.

5.4.4 Animationen Anzeigen

In den vorherigen Schritten haben wir eine `.h`-Datei erhalten, die eine Reihe von Frames zusammen mit ihren Dauern speichert. Als Nächstes lassen Sie uns diese auf der LED-Matrix anzeigen.

1. Erstellen Sie zuerst einen Sketch. Sie können entweder die Datei `04-led_matrix.ino` öffnen, die sich unter dem Pfad `elite-explorer-kit-main\r4_new_feature\04-led_matrix` befindet, oder diesen Code in die **Arduino IDE** kopieren.
2. Wenn Sie den Code vom Pfad `elite-explorer-kit-main\r4_new_feature\04-led_matrix` verwenden, finden Sie in der Arduino IDE einen Tab namens `animation.h`. Öffnen Sie diesen und ersetzen Sie den vorhandenen Code durch die `.h`-Datei, die Sie von der Webseite erhalten haben.



3. Wenn Sie Ihren eigenen Sketch erstellt haben, müssen Sie die von der Webseite erhaltene .h-Datei in dasselbe Verzeichnis Ihres Sketches kopieren.
4. Nachdem Sie Ihren bevorzugten Code in der Arduino IDE eingerichtet und auf Ihr Arduino UNO R4 WiFi hochgeladen haben, sollte Ihre LED-Matrix nun das von Ihnen definierte Muster anzeigen.

Herzlichen Glückwunsch! Sie haben erfolgreich die 12x8 LED-Matrix Ihres Arduino UNO R4 WiFi programmiert!

Referenz

-

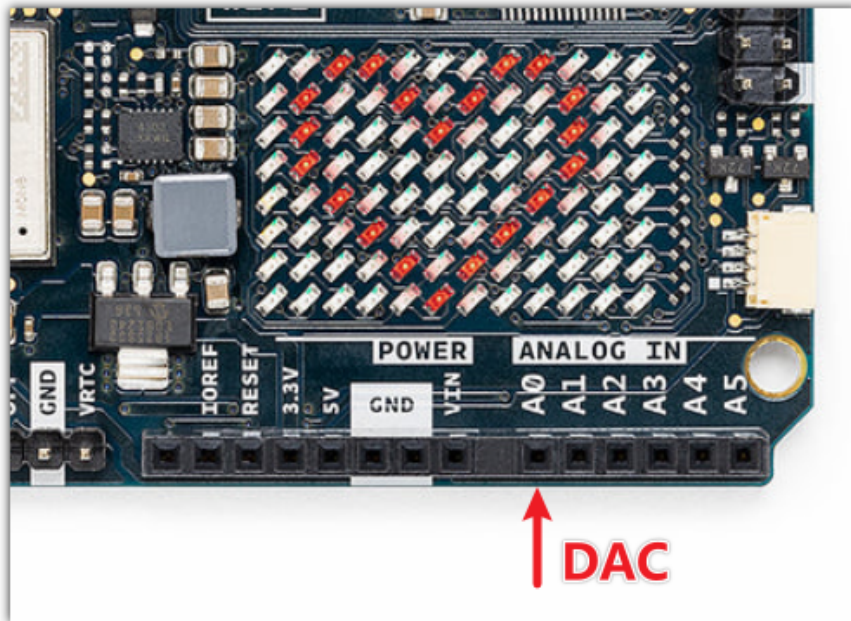
Weitere Projekte

- [SPIEL - Snake](#) (Spaßprojekt)

5.5 Digital-Analog-Wandler (DAC)

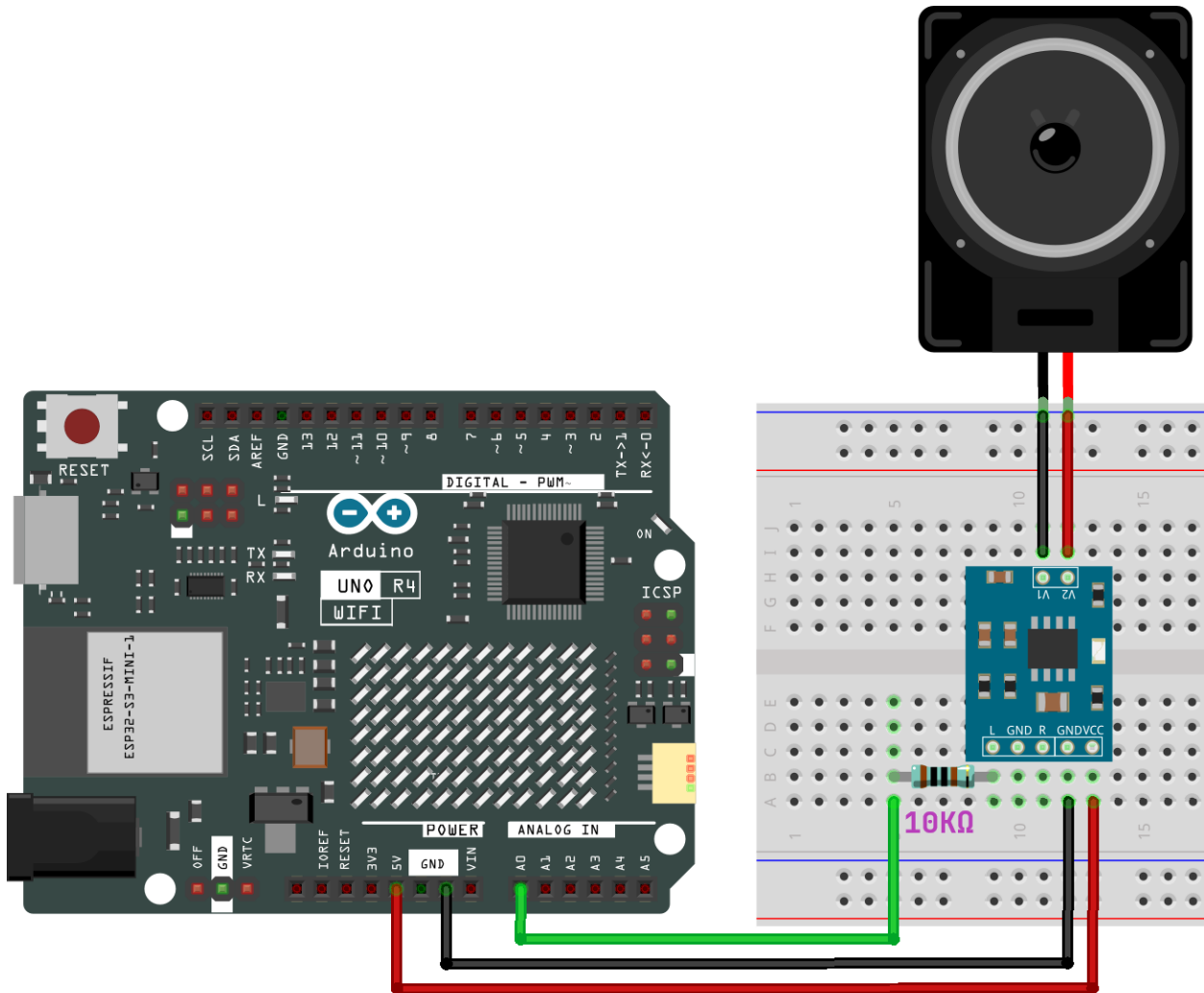
Das Arduino Uno R4 WiFi ist mit einem integrierten DAC (Digital-Analog-Wandler) ausgestattet. Ein DAC ist entscheidend für die Umwandlung digitaler Signale in ihre analogen Gegenstücke, eine Funktionalität, die besonders wichtig in Anwendungen wie der Audiobearbeitung, der Erzeugung analoger Signale und anderen Szenarien ist, die präzise analoge Ausgaben erfordern.

Der DAC auf dem Uno R4 WiFi bietet eine Auflösung von bis zu 12 Bit und liefert echte analoge Ausgabefähigkeiten, die die Leistung von PWM-Pins übertreffen.



5.5.1 Musik mit DAC abspielen

Schaltplan



Code Hochladen

Öffnen Sie die Datei `05-dac.ino`, die sich unter `elite-explorer-kit-main\r4_new_feature\05-dac` befindet, oder fügen Sie den folgenden Code in Ihre Arduino IDE ein.

Bemerkung: Bitte platzieren Sie die Datei `pitches.h` im selben Verzeichnis wie den Code, um eine ordnungsgemäße Funktion zu gewährleisten.

Dieses Projekt nutzt das Arduino und den DAC (Digital-Analog-Wandler), um die ikonische Melodie von Super Mario Bros zu spielen. Es verwendet eine Bibliothek namens `analogWave` zur Erzeugung von Sinuswellen und eine weitere Bibliothek, `pitches.h`, zur Definition der Notenfrequenzen.

- `melody[]`: Dieses Array enthält die zu spielenden Noten zusammen mit ihren Dauern. Noten werden durch vordefinierte Töne (z.B. `NOTE_E5`) dargestellt, und Dauern werden in Form von Schlägen repräsentiert (z.B. 4 für eine Viertelnote). Sie können versuchen, Ihre eigene Melodie zu komponieren, indem Sie die Noten und Dauern im Array `melody[]` ändern. Wenn Sie interessiert sind, gibt es ein Repository auf GitHub (<https://github.com/tonyduffy/arduino-melody>), das Arduino-Code für verschiedene Lieder bereitstellt. Obwohl deren Ansatz von diesem Projekt abweichen mag, können Sie sich an ihren Noten und Dauern orientieren. (Ersetzen Sie einfach das `melody[]` im entsprechenden Track durch den Code in diesem Projekt.)
- `tempo`: Das Tempo für dieses Projekt ist auf 200 BPM (Beats Per Minute) eingestellt, das zur Berechnung der

Dauer jeder Note verwendet wird. Die Änderung dieses Werts verändert die Geschwindigkeit der Songaufführung.

- **Sinuswellengenerator:** Die Funktion `sine` der Bibliothek `analogWave` initialisiert einen 10 Hz Sinuswellengenerator, der zur Ausgabe der Noten über den DAC verwendet wird.
- **Notendauer:** Basierend auf dem eingestellten Tempo und der Schlagzahl für jede Note wird die Dauer jeder Note berechnet.
- **Spielen und Pause:** Jede Note wird für 85% ihrer berechneten Dauer gespielt, gefolgt von einer 15% Pause, um zwischen den Noten zu unterscheiden.
- **Schleife:** Nachdem die Melodie abgespielt wurde, setzt der Code automatisch zurück und beginnt erneut zu spielen.

Dies ist ein Beispiel, das demonstriert, wie man Arduino und externe Hardware (DAC) zur Musikgenerierung verwendet. Es zeigt auch, wie man Arrays und Schleifen verwendet, um die Logik der Musikwiedergabe zu vereinfachen.

Referenz

-

5.6 USB HID

Das Arduino Uno R4 WiFi ist nicht nur ein leistungsfähiges Entwicklungsboard; es bietet auch eingebaute Unterstützung für Human Interface Devices (HID). Dies ermöglicht es Ihnen, das Board zur Emulation von Geräten wie Mäusen und Tastaturen zu verwenden und so Ihren Projekten eine neue Ebene der Interaktivität hinzuzufügen.

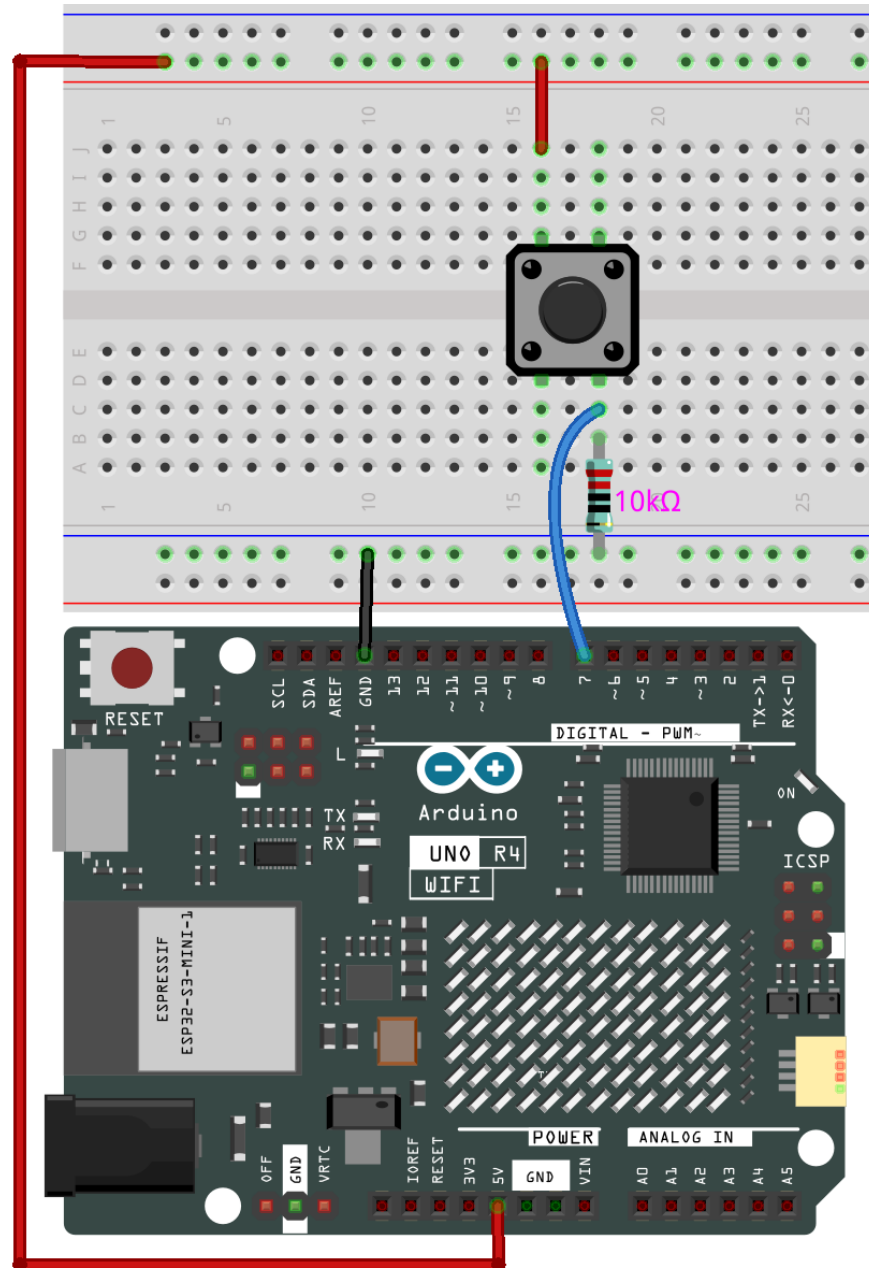
HID, oder Human Interface Devices, sind eine Kategorie von Computergeräten, die für die direkte Interaktion mit Menschen konzipiert sind, typischerweise zu Eingabezwecken. Zu dieser Kategorie gehören Geräte wie Tastaturen, Mäuse und Spielcontroller. Mit dem Arduino Uno R4 WiFi können Sie diese Geräte emulieren und so eine Vielzahl von Möglichkeiten für DIY-Projekte freischalten.

5.6.1 Maussteuerung

Die Steuerung einer Maus mit dem Arduino Uno R4 WiFi ist einfach. Mit dem Befehl `Mouse.move(x,y)` können Sie die Mausbewegung leicht steuern. Bei der Aktualisierung der Cursorposition ist sie immer relativ zur vorherigen Position des Cursors.

Hier ist ein einfaches Beispiel, das die Steuerung des Mauszeigers mit einem Knopf demonstriert.

Schaltplan



Code Hochladen

Öffnen Sie die Datei `06-hid_mouse.ino`, die sich unter `elite-explorer-kit-main\r4_new_feature\06-hid_mouse` befindet, oder fügen Sie den folgenden Code in Ihre Arduino IDE ein.

Warnung: Wenn Sie den Befehl `Mouse.move()` verwenden, übernimmt das Arduino die Kontrolle über die Maus Ihres Computers! Um sicherzustellen, dass Sie die Kontrolle über Ihren Computer nicht verlieren, während Sie einen Sketch mit dieser Funktion ausführen, richten Sie ein zuverlässiges Steuerungssystem ein, bevor Sie `Mouse.move()` aufrufen. Dieser Sketch enthält einen Druckknopf, um die Maus zu aktivieren, sodass sie erst nach Betätigung des Knopfes funktioniert.

Warnung: Aufgrund der Multi-Prozessor-Architektur des UNO R4 WiFi-Boards können beim Hochladen von Code, der HID-Funktionen verwendet, Fehler wie „No device found on...“ auftreten.

Um in solchen Fällen hochzuladen, folgen Sie diesen Schritten:

1. Drücken Sie schnell zweimal hintereinander die „RESET“-Taste auf dem Board. Die mit „L“ markierte LED sollte zu pulsieren beginnen.
2. Wählen Sie aus dem Arduino-IDE-Menü den Port des Boards aus. Der Port kann sich nach dem Reset ändern, stellen Sie also sicher, dass er korrekt ausgewählt ist.

Neben der Steuerung der Mausbewegung können Sie auch Mausclicks verwalten. Für weitere Informationen siehe .

5.6.2 Tastatursteuerung

Das Arduino Uno R4 WiFi bietet auch Tastaturemulationsfähigkeiten. Es ermöglicht Ihnen, nicht nur einzelne Tastenanschläge zu senden, sondern auch komplexe Tastenkombinationen auszuführen.

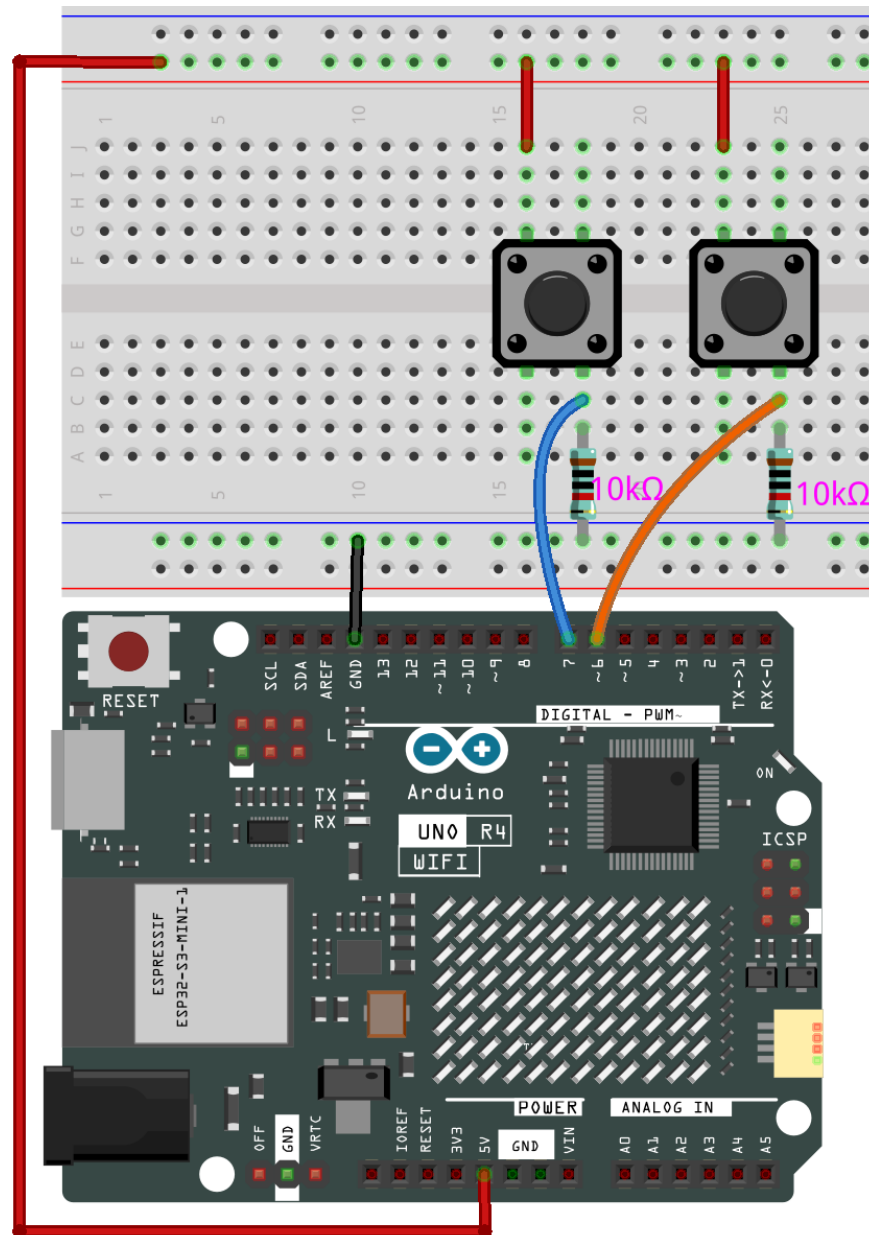
Warnung: Wenn Sie den Befehl `Keyboard.print()` verwenden, übernimmt das Arduino die Kontrolle über die Tastatur Ihres Computers! Um sicherzustellen, dass Sie die Kontrolle über Ihren Computer nicht verlieren, während Sie einen Sketch mit dieser Funktion ausführen, richten Sie ein zuverlässiges Steuerungssystem ein, bevor Sie `Keyboard.print()` aufrufen. Dieser Sketch enthält einen Druckknopf, um die Tastatur zu aktivieren, sodass sie erst nach Betätigung des Knopfes funktioniert.

Beispielcode für das Senden von Tastenkombinationen

In diesem Beispiel ist das Arduino Uno R4 WiFi so konfiguriert, dass es zwei häufig verwendete Tastenkombinationen emuliert: „Strg+C“ zum Kopieren und „Strg+V“ zum Einfügen. Zwei physische Knöpfe, die mit dem Arduino verbunden sind, dienen als Auslöser. Der Knopf, der mit Pin 7 verbunden ist, initiiert die Kopieraktion, während der mit Pin 8 verbundene Knopf das Einfügen auslöst.

Bei Betätigung eines der Knöpfe verwendet das Arduino die Funktionen `Keyboard.press()` und `Keyboard.releaseAll()`, um die jeweiligen Tastenkombinationen zu imitieren. Dieses Beispiel veranschaulicht, wie Sie eine dedizierte Hardware-Schnittstelle für spezifische Aufgaben entwerfen können, um repetitive Aktionen ohne Tastatureinsatz zu erleichtern. Dies könnte besonders vorteilhaft in Arbeitsumgebungen sein, die schnelle Datenmanipulation erfordern, oder in Zugänglichkeitseinrichtungen, die von vereinfachten Steuerungen profitieren.

Schaltplan



Code Hochladen

Öffnen Sie die Datei `06-hid_keyboard.ino`, die sich unter `elite-explorer-kit-main\r4_new_feature\06-hid_keyboard` befindet, oder fügen Sie den folgenden Code in Ihre Arduino IDE ein.

5.6.3 Vorsichtsmaßnahmen und Tipps

1. **Hinweis zu Maus- und Tastaturbibliotheken:** Wenn entweder die Maus- oder die Tastaturbibliothek kontinuierlich läuft, könnte dies die Programmierung Ihres Boards beeinträchtigen. Funktionen wie `Mouse.move()` und `Keyboard.print()` übernehmen die Kontrolle über Ihren angeschlossenen Computer und sollten nur aufgerufen werden, wenn Sie bereit sind, sie zu verwalten. Es wird empfohlen, ein Steuerungssystem, wie einen physischen Schalter oder spezifische Eingabesteuerungen, zu verwenden, um diese Funktionalität umzuschalten.
2. **Wenn Sie Probleme beim Hochladen des Codes haben:** Aufgrund der Multi-Prozessor-Architektur des UNO R4 WiFi-Boards können beim Hochladen von Code, der HID-Funktionalitäten verwendet, Fehler wie „Kein Gerät gefunden auf...“ auftreten.

Um unter solchen Umständen hochzuladen, folgen Sie diesen Schritten:

1. Drücken Sie schnell zweimal hintereinander die „RESET“-Taste auf dem Board. Die mit „L“ markierte LED sollte zu pulsieren beginnen.
2. Wählen Sie aus dem Arduino-IDE-Menü den Port des Boards aus. Der Port kann sich nach dem Reset ändern, stellen Sie also sicher, dass er korrekt ausgewählt ist.

Referenz

-
-
-
- **Wi-Fi®:** Bietet drahtlose Konnektivität, ideal für verschiedene IoT-Projekte, ermöglicht durch das ESP32-S3-Modul.
- **Bluetooth®:** Bietet Kurzstrecken-Drahtloskommunikation zwischen Geräten, ebenfalls angetrieben durch das ESP32-S3-Modul.
- **Eingebaute Echtzeituhr (RTC):** Ideal für zeitkritische Anwendungen. Beinhaltet zusätzliche Pins für batteriebetriebene Nutzung und einen „OFF“-Pin, um das Board auszuschalten, während die RTC weiterläuft.
- **12x8 LED-Matrix:** Eine einfache Möglichkeit, Daten anzuzeigen oder Animationen zu erstellen.
- **DAC-Kanal:** Erzielen Sie präzise analoge Ausgaben, perfekt für Audioprojekte.
- **HID-Unterstützung:** Simulieren Sie eine Maus oder Tastatur über USB mit eingebauter HID-Unterstützung.
- **CAN-Protokoll-Unterstützung:** Erweitern Sie Ihre Reichweite in automotive und industrielle Anwendungen.

In diesem Kapitel finden Sie einige unterhaltsame Projekte. Diese Projekte beinhalten den Einsatz mehrerer elektronischer Komponenten und veranschaulichen die grundlegende Logik, wie die meisten Programme mit der Realität interagieren.

6.1 Willkommen

In diesem Projekt verwenden wir einen PIR-Sensor, um menschliche Anwesenheit zu erkennen, und einen Lautsprecher, um eine Türklingel zu simulieren, ähnlich den Eingangstürklingeln in Convenience-Stores. Wenn ein Fußgänger in den Bereich des PIR-Sensors kommt, wird der Lautsprecher klingeln und damit eine Türklingel nachahmen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

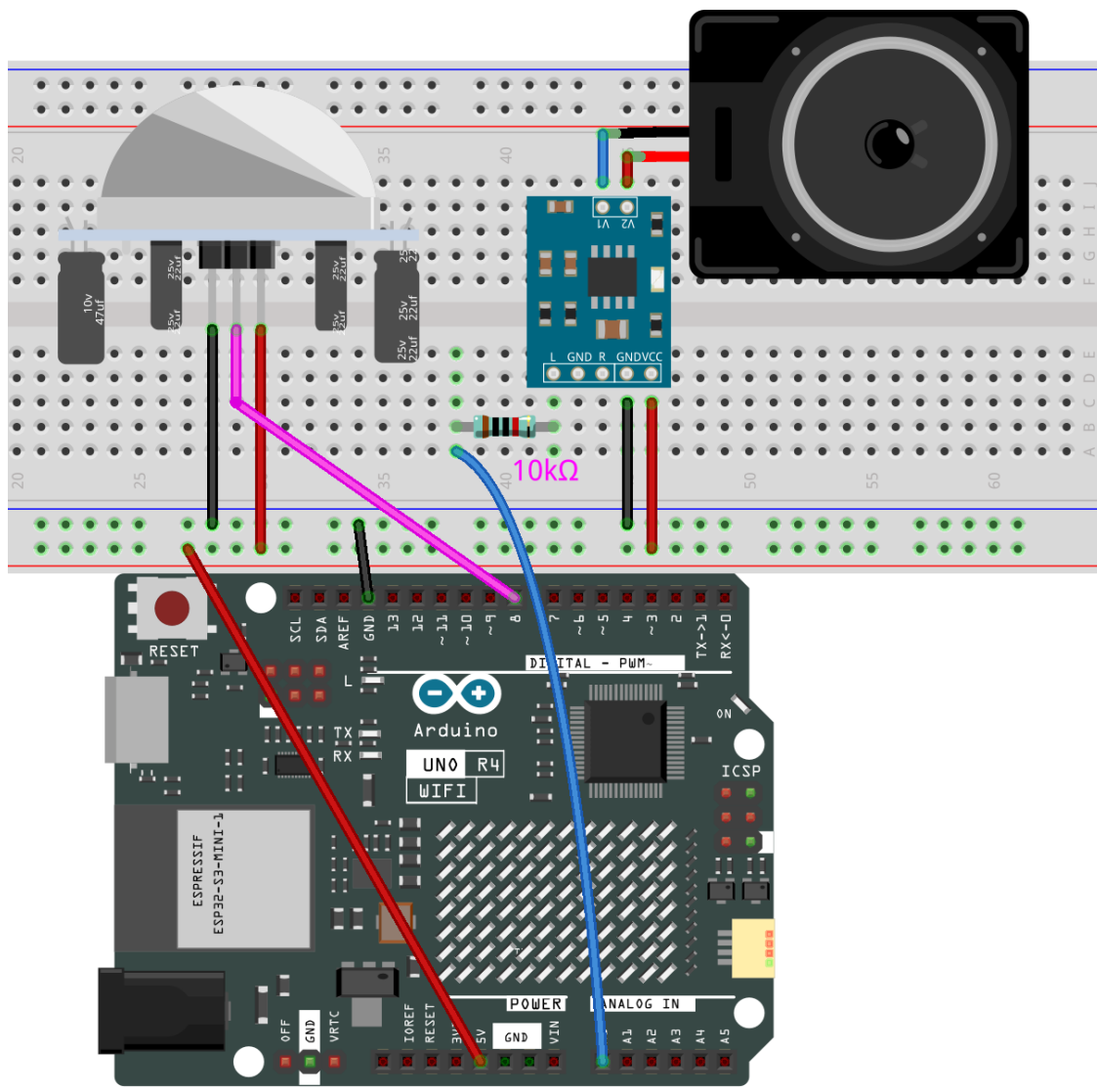
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

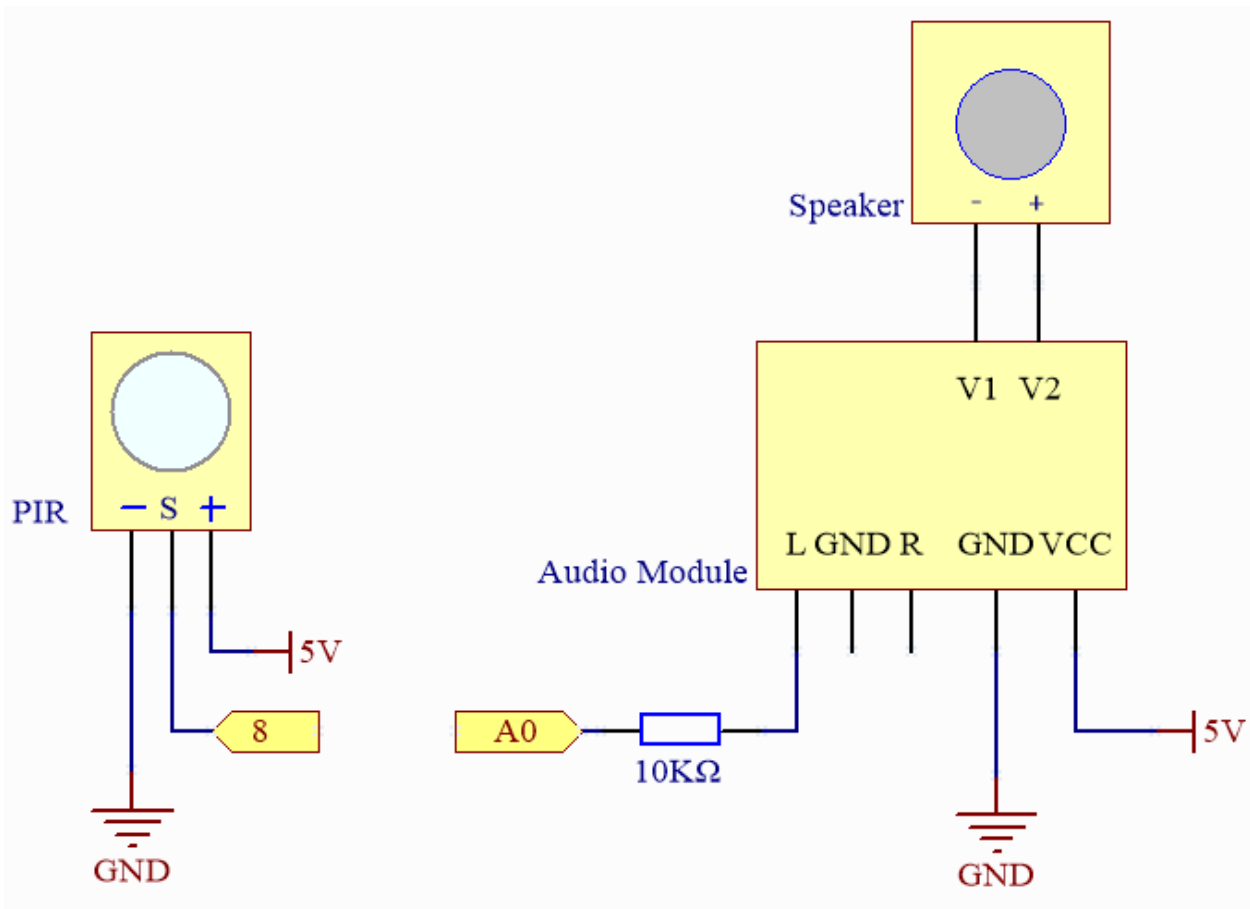
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>PIR-Bewegungssensormodul</i>	
<i>Audio-Modul und Lautsprecher</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `01_welcome.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\01_welcome` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert das?

Hier ist eine schrittweise Erklärung des Codes:

1. Headerdateien einbinden:

Zwei Headerdateien, `analogWave.h` und `pitches.h`, einbinden. Die Datei `analogWave.h` enthält die Definition der Klasse `analogWave`, während `pitches.h` die Definitionen musikalischer Noten enthält.

2. Objekte instanziiieren und Konstanten definieren:

Erstellen Sie ein `wave`-Objekt mit der Klasse `analogWave` und definieren Sie `PIR_PIN` als 2, welcher der an den PIR-Sensor angeschlossene Pin ist.

3. Melodie-Array:

Das `melody`-Array definiert eine musikalische Melodie, wobei jeder Note eine Zahl folgt, die ihre Dauer repräsentiert. Negative Zahlen stellen punktierte Noten dar (die Dauer um 50% erhöhen).

4. Globale Variablen:

Definieren Sie einige globale Variablen, um Daten zwischen Funktionen zu teilen.

5. `setup()`:

Initialisieren Sie PIR_PIN als Eingang und stellen Sie die Frequenz der Sinuswelle auf 10 Hz mit `wave.sine(10)` ein.

6. `loop()`:

Überwachen Sie kontinuierlich den Wert des PIR-Sensors. Wenn eine menschliche Anwesenheit erkannt wird (`pirValue` ist HIGH), rufen Sie die Funktion `playMelody()` auf, um die Melodie zu spielen, und warten Sie 10 Sekunden, um das wiederholte Abspielen der Melodie zu verhindern.

7. `playMelody()`:

Diese Funktion berechnet die Dauer jeder Note basierend auf den Daten im `melody`-Array und spielt die entsprechende Note. Zwischen den Noten gibt es eine kurze Pause. Die Funktion setzt die Frequenz der Wellenform mit `wave.freq()` und steuert die Dauer der Noten und Pausen zwischen den Noten mit der Funktion `delay()`.

Hinweis: Stellen Sie sicher, dass die Headerdatei `pitches.h` tatsächlich existiert, bevor Sie diesen Code ausführen.

6.2 Obstklavier

Dieses Projekt ist ein einfaches Obstklavier, das Eingaben von einem MPR121-Touchsensor liest und Musik über einen DAC abspielt. Mit anderen Worten, wir haben Früchte in eine Tastatur verwandelt, die es Ihnen ermöglicht, Musik zu spielen, indem Sie sie einfach berühren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

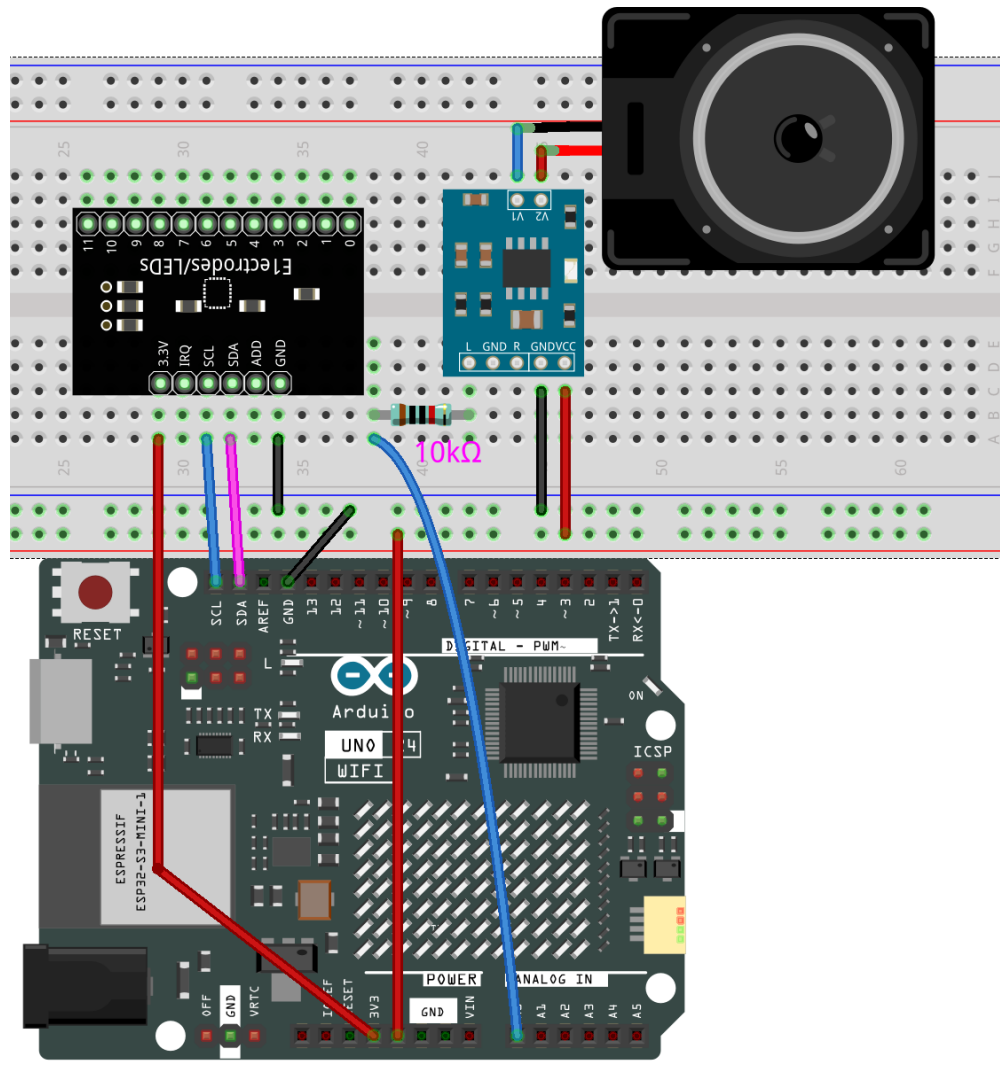
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

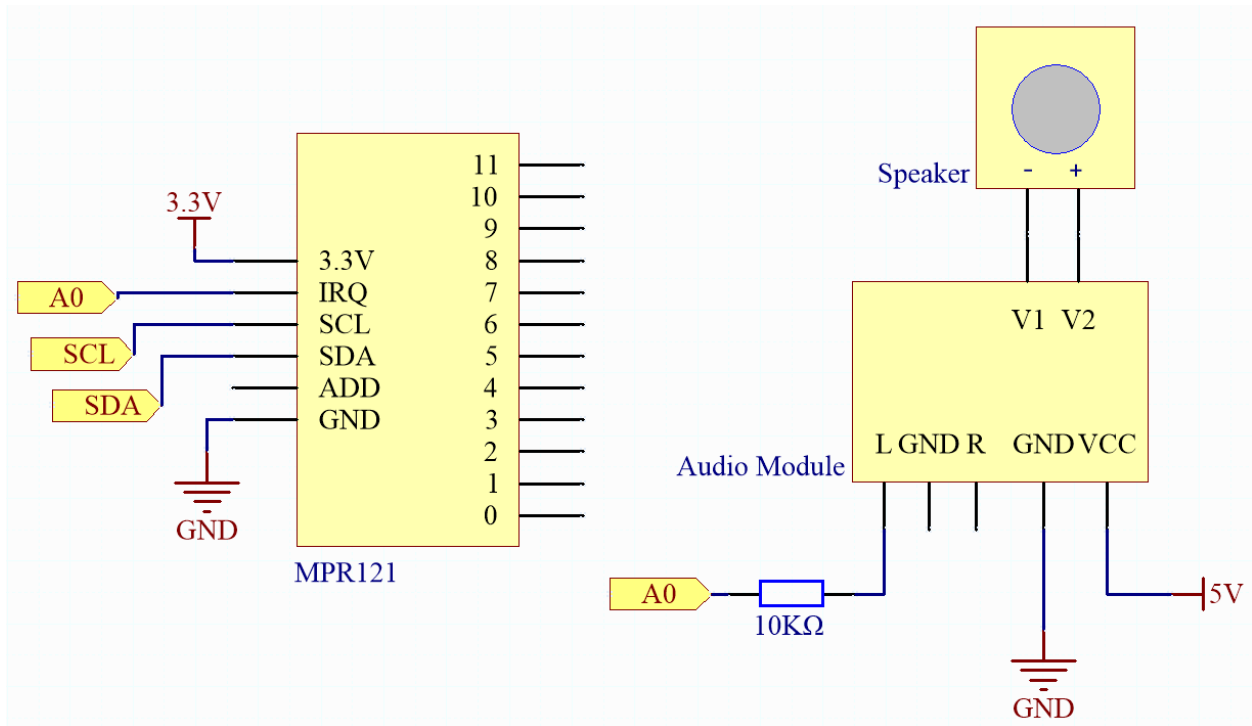
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>MPR121</i>	-
<i>Audio-Modul und Lautsprecher</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei 02_fruit_piano.ino direkt unter dem Pfad elite-explorer-kit-main\fun_project\02_fruit_piano öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager und suchen Sie nach „Adafruit MPR121“ und installieren Sie diese.

Wie funktioniert das?

Hier ist eine schrittweise Erklärung des Codes:

1. Bibliotheks- und Objektinitialisierung:

Importieren Sie die notwendigen Bibliotheken: Wire-Bibliothek (für I2C-Kommunikation), Adafruit_MPR121-Bibliothek (zur Steuerung des MPR121), analogWave-Bibliothek (zur Erzeugung analoger Wellenformen) und pitches.h (definiert die Frequenzen der Noten). Erstellen Sie Instanzen der Objekte Adafruit_MPR121 und analogWave. Definieren Sie ein Noten-Array, um die Note zu speichern, die jedem Berührungskanal entspricht.

2. setup():

Initialisieren Sie die serielle Kommunikation und warten Sie, bis sie startet. Überprüfen und initialisieren Sie den MPR121; falls nicht gefunden, geben Sie eine Fehlermeldung auf dem seriellen Monitor aus und stoppen Sie die Ausführung. Initialisieren Sie das analogWave-Objekt und setzen Sie die anfängliche Sinuswellenfrequenz auf 10 Hz.

3. loop():

Lesen Sie die aktuell berührten Kanäle des MPR121. Durchlaufen Sie alle Kanäle, überprüfen Sie, welcher berührt wird, und spielen Sie die entsprechende Note. Fügen Sie eine kleine Verzögerung zwischen jeder Iteration hinzu.

4. Note abspielen `playNote()`:

Die Funktion `playNote` nimmt einen `note`-Parameter entgegen und setzt die DAC-Frequenz, um die entsprechende Note zu spielen. Verzögern Sie für einen Zeitraum, um die Note zu spielen. Stoppen Sie das Abspielen der Note.

6.3 HueDial

Dieses Beispiel steuert die Farbe einer RGB-LED basierend auf der Position eines Drehknopfs. Verschiedene Positionen des Knopfs entsprechen verschiedenen HUE-Werten, die dann in RGB-Farbwerte übersetzt werden, was zu einer Farbänderung der RGB-LED führt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

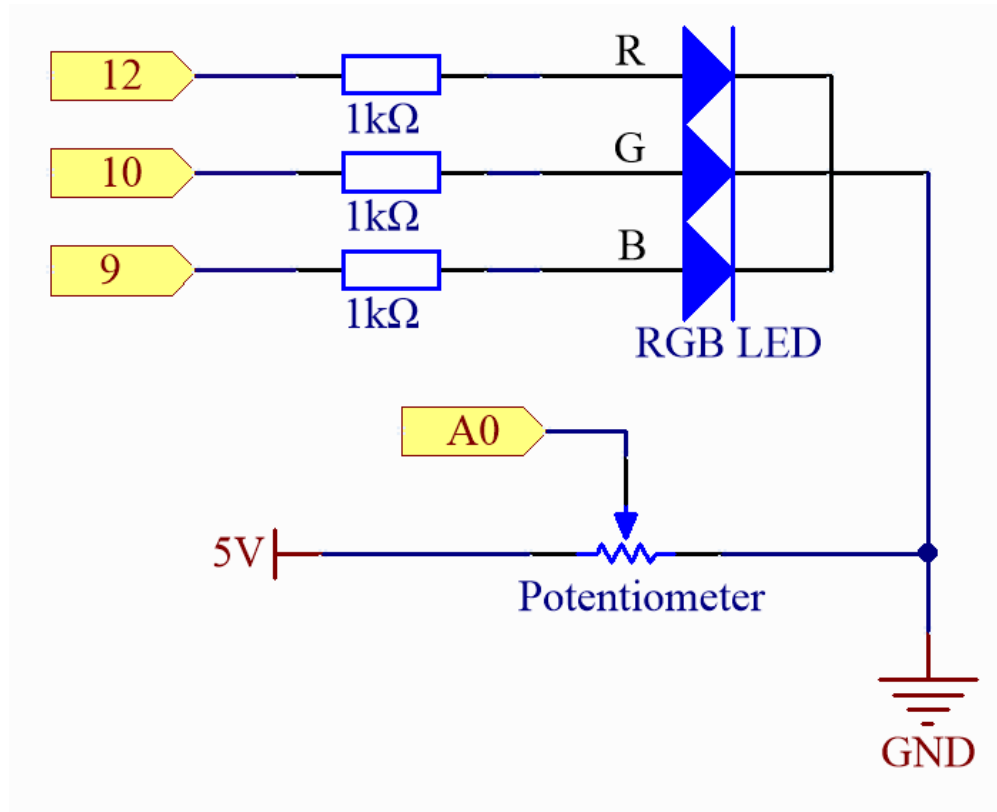
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>RGB-LED</i>	
<i>Potentiometer</i>	

Verdrahtung





Code

Bemerkung:

- Sie können die Datei `03_huedial.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\03_huedial` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert das?

Hier ist eine detaillierte Erklärung des Codes:

1. Globale Variablendefinitionen:

`redPin`, `greenPin` und `bluePin`: Diese definieren die PWM-Pins, die jeweils mit den roten, grünen und blauen LEDs verbunden sind. `KNOB_PIN`: Definiert den analogen Eingangspin, der mit dem Drehknopf verbunden ist.

2. `setup()`:

Stellen Sie die Pins für die RGB-LED als Ausgang ein. Standardmäßig sind analoge Pins als Eingänge gesetzt, daher ist es nicht notwendig, den Eingangsmodus für den Pin des Knopfes einzustellen.

3. `loop()`:

Lesen Sie den Wert des Drehknopfs. Dieser Wert reicht von 0 bis 1023. Normalisieren Sie den Wert des Knopfs auf einen Bereich von 0-1. Konvertieren Sie den normalisierten Wert in einen HUE-Wert im Bereich von 0-360. Konvertieren Sie den HUE-Wert in RGB-Werte. Aktualisieren Sie die Farbe der LED mit diesen RGB-Werten.

4. `setColor()`:

Setzen Sie die passenden PWM-Werte für jeden LED-Pin mit der Funktion `analogWrite()`, um die Farbe der RGB-LED einzustellen.

5. `HUEtoRGB()`:

Diese Funktion konvertiert HUE-Werte in RGB-Werte mit der HSL-zu-RGB-Konvertierungsmethode, konzentriert sich jedoch nur auf die HUE-Komponente und hält Sättigung und Helligkeit bei 100%. Der Algorithmus ist in 6 Stufen unterteilt, wobei jede 60 Grad abdeckt. Es berechnet RGB-Werte für jede HUE-Stufe und skaliert dann diese Werte auf einen Bereich von 0-255, was der erwarteten Spanne für die Funktion `analogWrite()` entspricht.

6.4 Lichtempfindliches Array

Dieses Programm wandelt die Messwerte eines lichtabhängigen Widerstands in eine entsprechende Anzahl beleuchteter LED-Lichter um und schafft so einen einfachen Indikator für die Helligkeit des Lichts.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

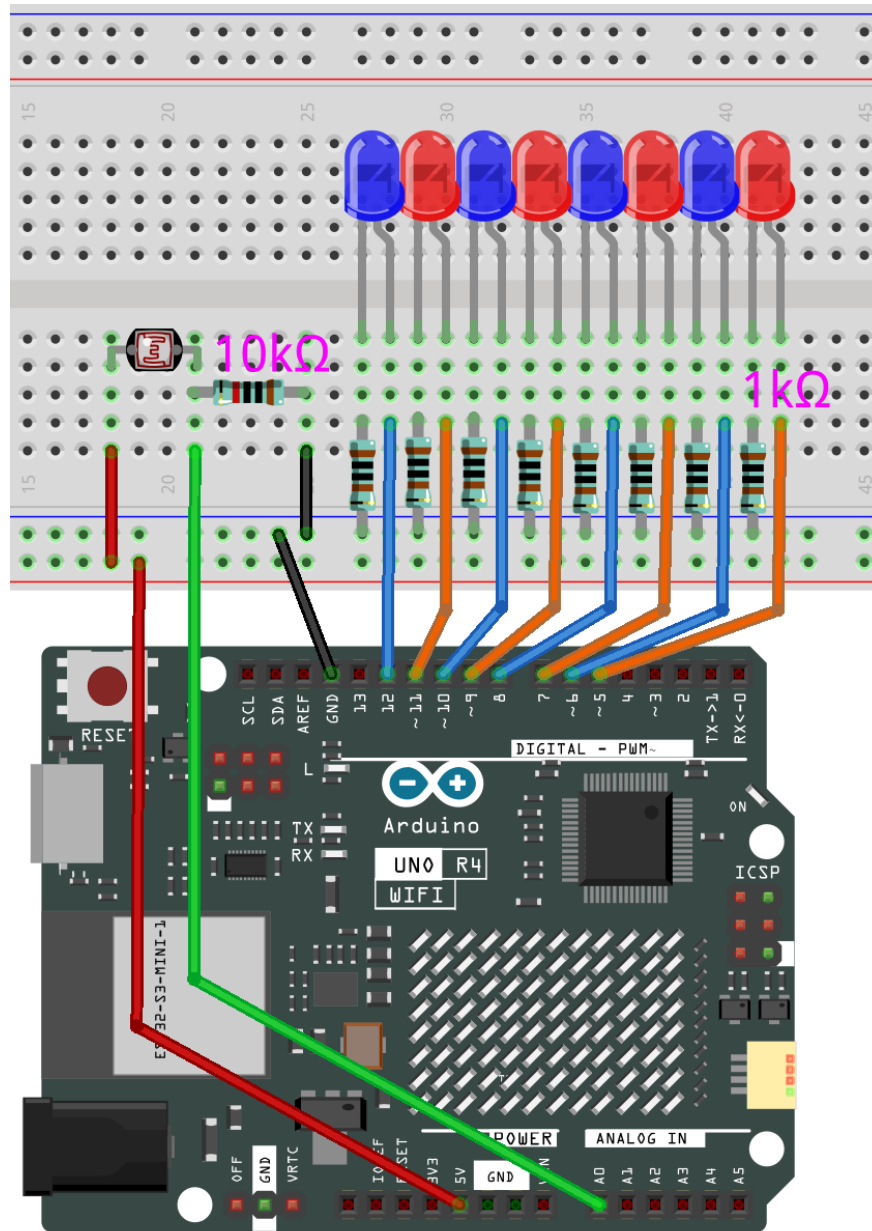
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

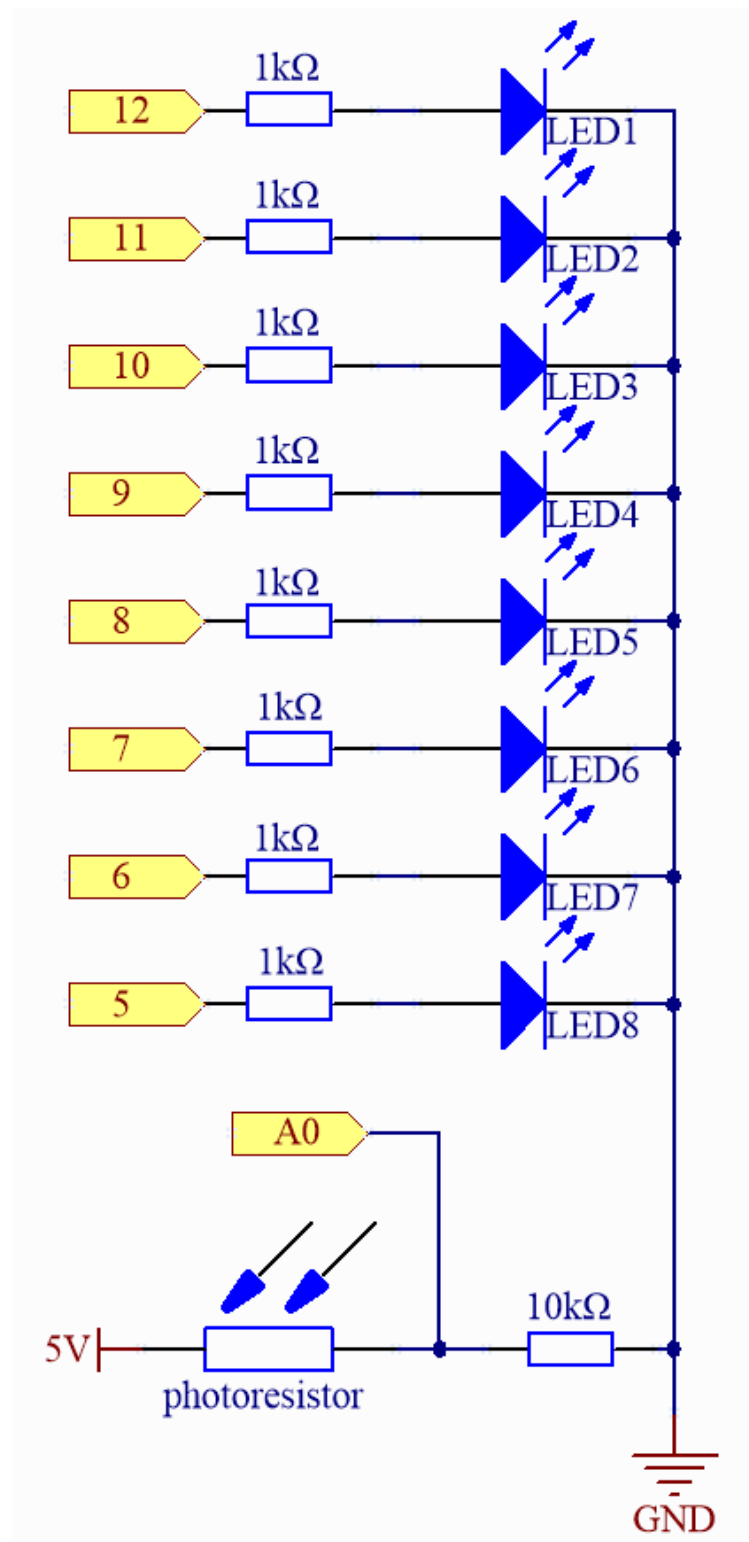
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Fotowiderstand</i>	

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `04_light_sensitive_array.ino` direkt unter dem Pfad `elite-explorer-kit-main\`

fun_project\04_light_sensitive_array öffnen.

- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert das?

Hier ist eine schrittweise Erklärung des Codes:

1. Konstanten- und Variablendefinitionen:

NbrLEDs: Definiert die Anwesenheit von 8 LEDs. **ledPins[]:** LEDs sind an die Arduino-Pins 5 bis 12 angeschlossen. **photoCellPin:** Der Fotowiderstand ist an den A0-Pin des Arduino angeschlossen. **sensorValue:** Diese Variable speichert den vom Fotowiderstand gelesenen Wert. **ledLevel:** Diese Variable speichert die Anzahl der LEDs basierend auf der Umwandlung des sensorValue.

2. setup():

Konfiguriert die Pins 5 bis 12 als Ausgang, um die LEDs anzusteuern.

3. loop():

Liest den analogen Wert des Fotowiderstands vom Pin A0, typischerweise im Bereich von 0 bis 1023. Verwendet die Funktion map, um den Wert des Fotowiderstands von 300-1023 auf den Bereich 0-8 abzubilden. Das bedeutet, dass, wenn die Messung des lichtabhängigen Widerstands 300 beträgt, keine LEDs leuchten werden; wenn die Messung 1023 oder höher ist, werden alle 8 LEDs leuchten.

Die nachfolgende for-Schleife überprüft jede LED. Wenn ihr Index kleiner als ledLevel ist, wird die LED eingeschaltet; andernfalls wird sie ausgeschaltet.

6.5 Digitaler Würfel

Dieser Code ist darauf ausgelegt, einen rollenden Würfel mit einem 74HC595-Schieberegister und einer 7-Segment-Digitalanzeige zu simulieren. Die Würfelwurfsimulation wird durch direktes Schütteln des Neigungsschalters aktiviert. Bei dieser Aktion durchläuft die digitale Anzeige zufällige Zahlen zwischen 1 und 6, was das Rollen eines Würfels simuliert. Nach einer kurzen Pause stoppt die Anzeige und zeigt eine zufällige Zahl an, die das Ergebnis des Würfelwurfs darstellt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

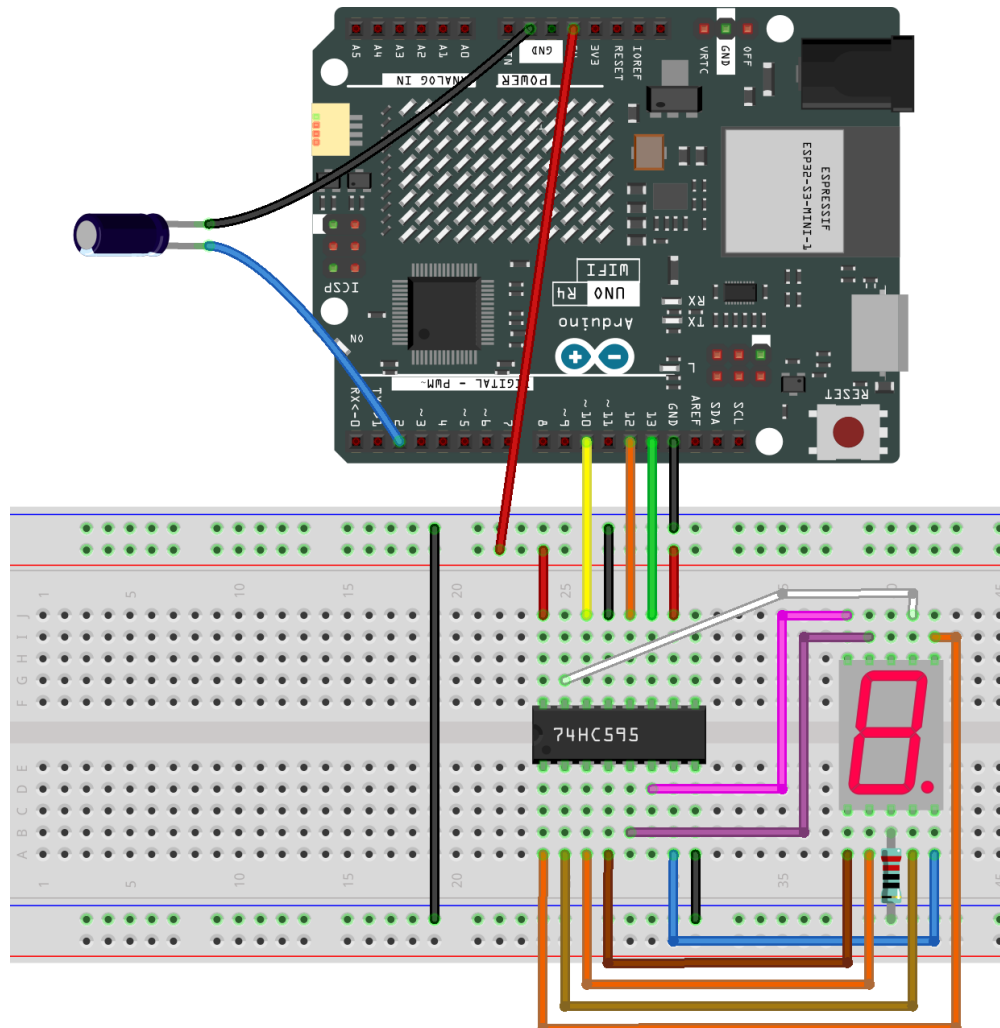
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

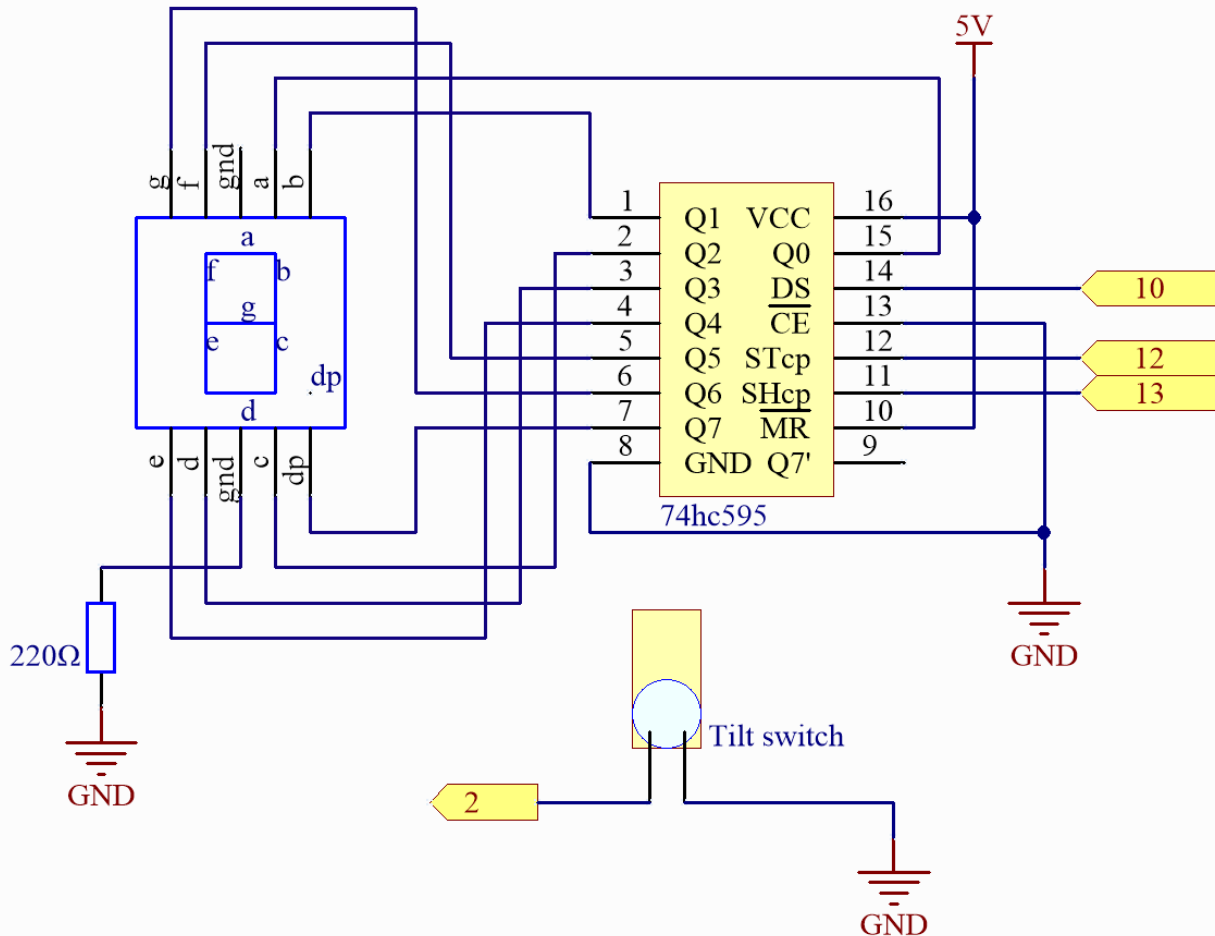
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>Neigungsschalter</i>	-
<i>74HC595</i>	
<i>7-Segment-Anzeige</i>	

Verkabelung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `05_digital_dice.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\05_digital_dice` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert das?

Hier ist eine detaillierte Erklärung des Codes:

1. Initialisierung von Variablen:

`dataPin`, `clockPin`, `latchPin`: Pins für den 74HC595. `buttonPin`: Der digitale Pin, an den der Knopf angeschlossen ist. `numbers[]`: Ein Array, um die Kodierung zu speichern, die Zahlen 1 bis 6 auf einer gemeinsamen Anoden-Digitalröhre darstellt.

2. Flüchtige Variablen:

`rolling`: Dies ist eine flüchtige Variable, die angibt, ob der Würfel gerade rollt. Sie wird als flüchtig deklariert, da sie sowohl in der Interrupt-Service-Routine als auch im Hauptprogramm verwendet wird.

3. `setup()`:

Setzen Sie die Modi für die relevanten Pins. Stellen Sie den Eingangsmodus für den Knopf unter Verwendung des internen Pull-Up-Widerstands ein. Weisen Sie dem Knopf einen Interrupt zu, der die Funktion `rollDice` aufruft, wenn sich der Zustand des Knopfes ändert.

4. `loop()`:

Es wird überprüft, ob `rolling` wahr ist. Wenn ja, wird weiterhin eine zufällige Zahl zwischen 1 und 6 angezeigt. Wenn der Knopf länger als 500 Millisekunden gedrückt wurde, stoppt das Rollen.

5. `rollDice()`:

Dies ist die Interrupt-Service-Routine für den Knopf. Es wird überprüft, ob der Knopf gedrückt ist (niedriges Niveau). Wenn ja, wird die aktuelle Zeit aufgezeichnet und das Rollen beginnt.

6. `displayNumber()`:

Diese Funktion zeigt eine Zahl auf der Digitalröhre an. Sie sendet die Zahl über den 74HC595-Schieberegister an die Digitalröhre.

6.6 Smarter Ventilator

Dieses Arduino-Projekt passt die Geschwindigkeit des Ventilators automatisch an, um die Temperatur in einem geeigneten Bereich zu halten. Zusätzlich können Benutzer über einen Knopf in den manuellen Modus wechseln und den Ventilator mit maximaler Geschwindigkeit betreiben.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

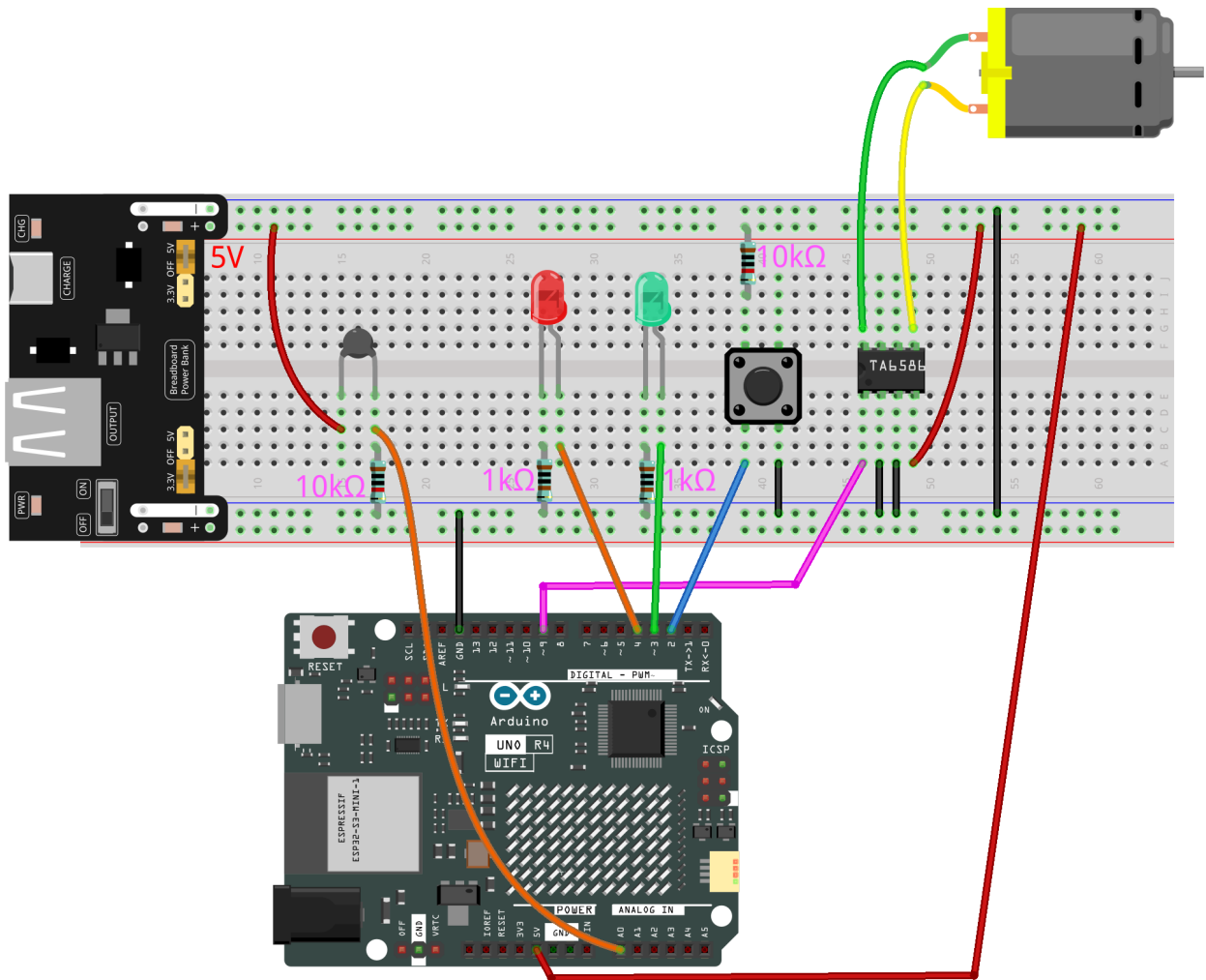
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

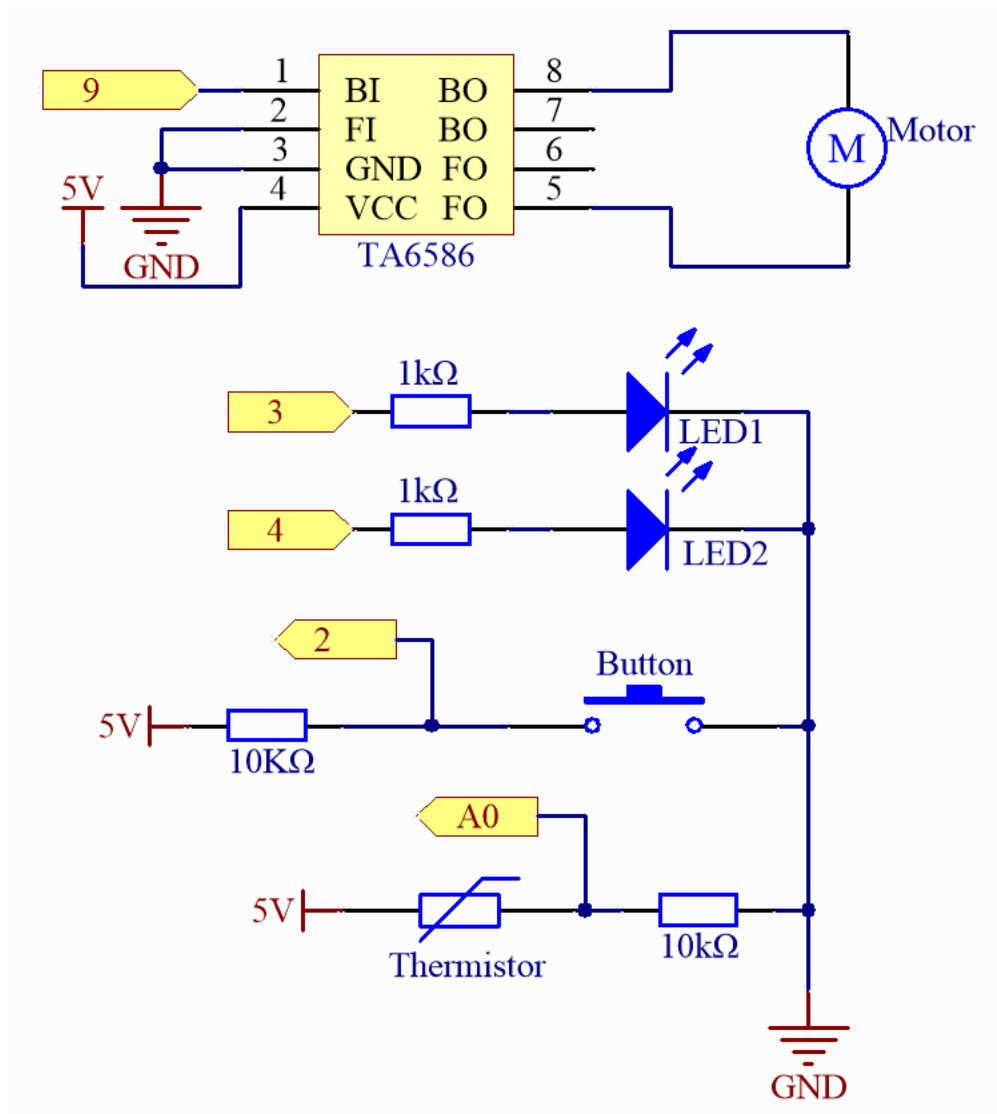
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Knopf</i>	
<i>Thermistor</i>	
<i>Gleichstrommotor</i>	
<i>TA6586 - Motorsteuerungs-Chip</i>	-
<i>Stromversorgungsmodul</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `06_smart_fan.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\06_smart_fan` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert des?

Hier ist eine schrittweise Erklärung des Codes:

1. Konstanten- und Variablendefinitionen:

Verwenden Sie `#define`, um die Pins für verschiedene Hardwareverbindungen zu definieren. `TEMP_THRESHOLD` ist als `25°C` definiert, was die Temperaturschwelle für den Start des Ventilators ist. `manualMode`: Eine boolesche Variable, die angibt, ob sich das Gerät im manuellen Modus befindet.

2. `setup()`:

Stellen Sie den Modus für relevante Pins ein (Ausgang, Eingang, Eingang mit Pull-up). Anfangs auf automatischen Modus eingestellt, sodass LED_AUTO leuchtet, während LED_MANUAL ausgeschaltet ist.

3. loop():

Überwachen Sie den Zustand des Knopfs. Wenn der Knopf gedrückt wird, wechselt er den Modus und ändert den Status der LED. Im manuellen Modus arbeitet der Ventilator mit maximaler Geschwindigkeit. Im automatischen Modus liest der Code zuerst den Spannungswert vom Temperatursensor und wandelt ihn in einen Temperaturwert um. Wenn die Temperatur die Schwelle überschreitet, wird die Geschwindigkeit des Ventilators anhand der Temperatur eingestellt.

4. voltageToTemperature():

Dies ist eine Hilfsfunktion, die verwendet wird, um den Spannungswert vom Temperatursensor in einen Temperaturwert (in Celsius) umzuwandeln. Die Funktion verwendet die Standardformel für einen Thermistor, um die Temperatur zu schätzen. Der Rückgabewert ist in Grad Celsius.

6.7 Smarter Mülleimer

Dies ist ein Arduino-Code, der entwickelt wurde, um einen intelligenten Mülleimer zu steuern. Wenn sich ein Objekt in einem Bereich von 20 Zentimetern vor dem Mülleimer befindet, öffnet sich der Deckel automatisch. Dieses Projekt nutzt einen SG90-Servomotor und einen HC-SR04-Ultraschall-Entfernungssensor.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

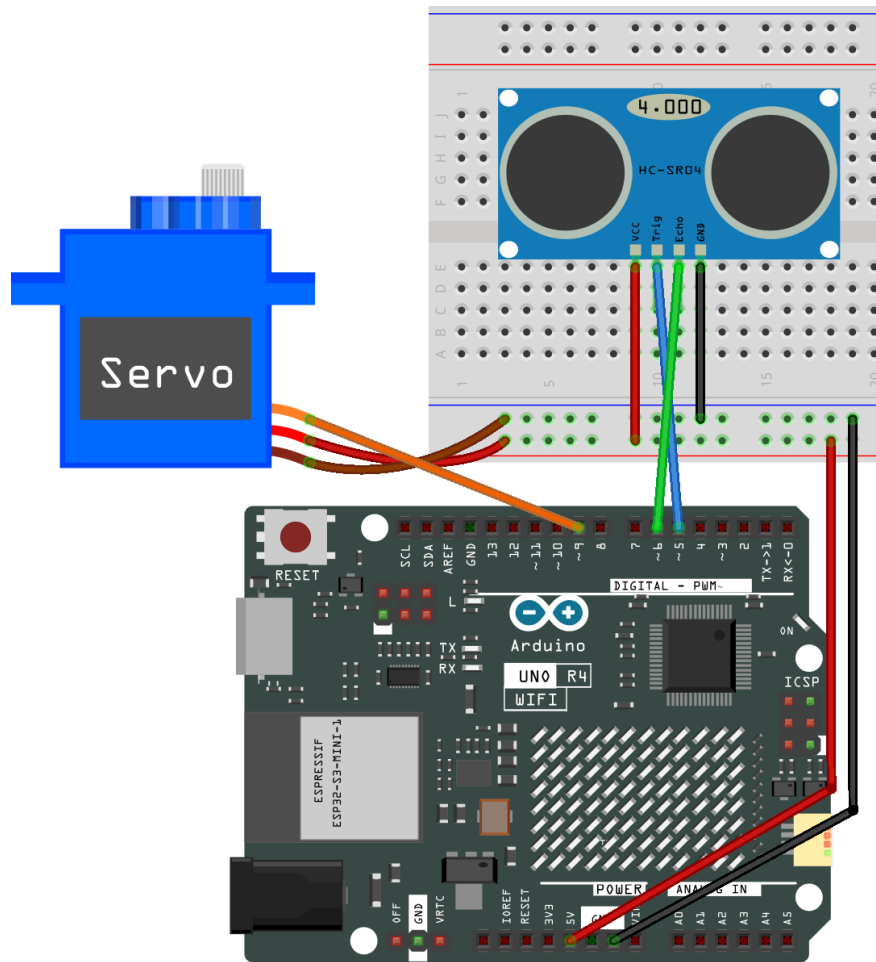
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

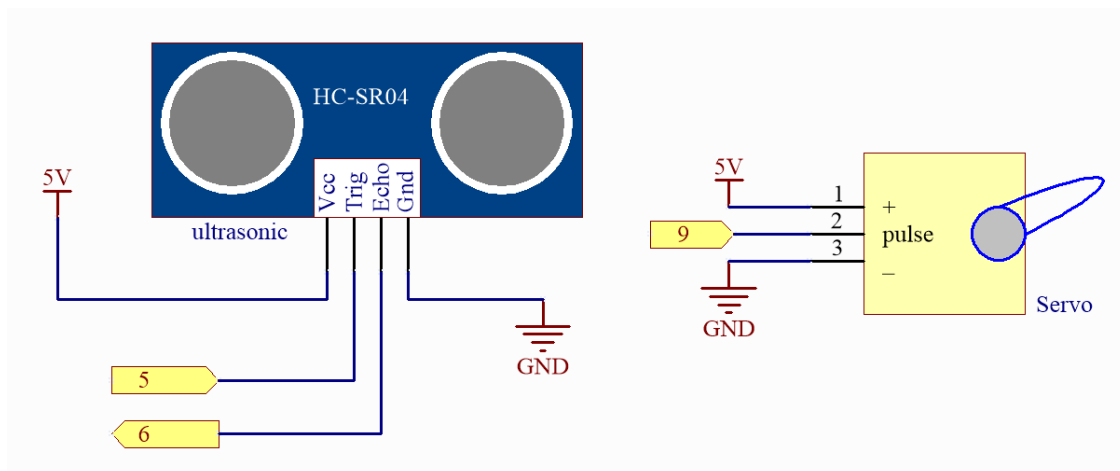
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Ultraschallmodul</i>	
<i>Servomotor</i>	

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei 07_smart_trash_can.ino direkt unter dem Pfad elite-explorer-kit-main\

`fun_project\07_smart_trash_can` öffnen.

- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert des?

Hier ist eine schrittweise Erklärung des Codes:

1. Bibliotheken importieren und Konstanten/Variablen definieren:

Die Bibliothek `Servo.h` wird importiert, um den SG90-Servomotor zu steuern. Parameter für den Servomotor, Ultraschallsensor und andere erforderliche Konstanten und Variablen werden definiert.

2. `setup()`:

Initialisieren Sie die serielle Kommunikation mit dem Computer mit einer Baudrate von 9600. Konfigurieren Sie die Trigger- und Echopins des Ultraschallsensors. Befestigen Sie den Servomotor an seinem Steuerpin und setzen Sie seine Anfangsposition auf den geschlossenen Winkel. Nachdem der Winkel eingestellt wurde, wird der Servomotor abgetrennt, um Energie zu sparen.

3. `loop()`:

Messen Sie die Entfernung dreimal und speichern Sie die Werte jeder Messung. Berechnen Sie die durchschnittliche Entfernung aus den drei Messungen. Wenn die durchschnittliche Entfernung weniger als oder gleich 20 Zentimetern (definierter Abstandsschwellenwert) beträgt, dreht sich der Servomotor auf den offenen Winkel (0 Grad). Andernfalls kehrt der Servomotor nach einer einsekündigen Verzögerung in die geschlossene Position (90 Grad) zurück. Der Servomotor wird abgetrennt, wenn er nicht verwendet wird, um Energie zu sparen.

4. `readDistance()`:

Senden Sie einen Impuls an den Triggerpin des Ultraschallsensors. Messen Sie die Impulsbreite des Echopins und berechnen Sie den Entfernungswert. Diese Berechnung verwendet die Schallgeschwindigkeit in der Luft, um die Entfernung basierend auf der Pulsdauer zu berechnen.

6.8 Pflanzenmonitor

Dieses Projekt bewässert Pflanzen automatisch, indem es eine Wasserpumpe aktiviert, wenn die Bodenfeuchtigkeit unter einen bestimmten Schwellenwert fällt. Zusätzlich werden Temperatur, Luftfeuchtigkeit und Bodenfeuchtigkeit auf einem LCD-Bildschirm angezeigt, um den Benutzern Einblicke in die Wachstums Umgebung der Pflanze zu geben.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

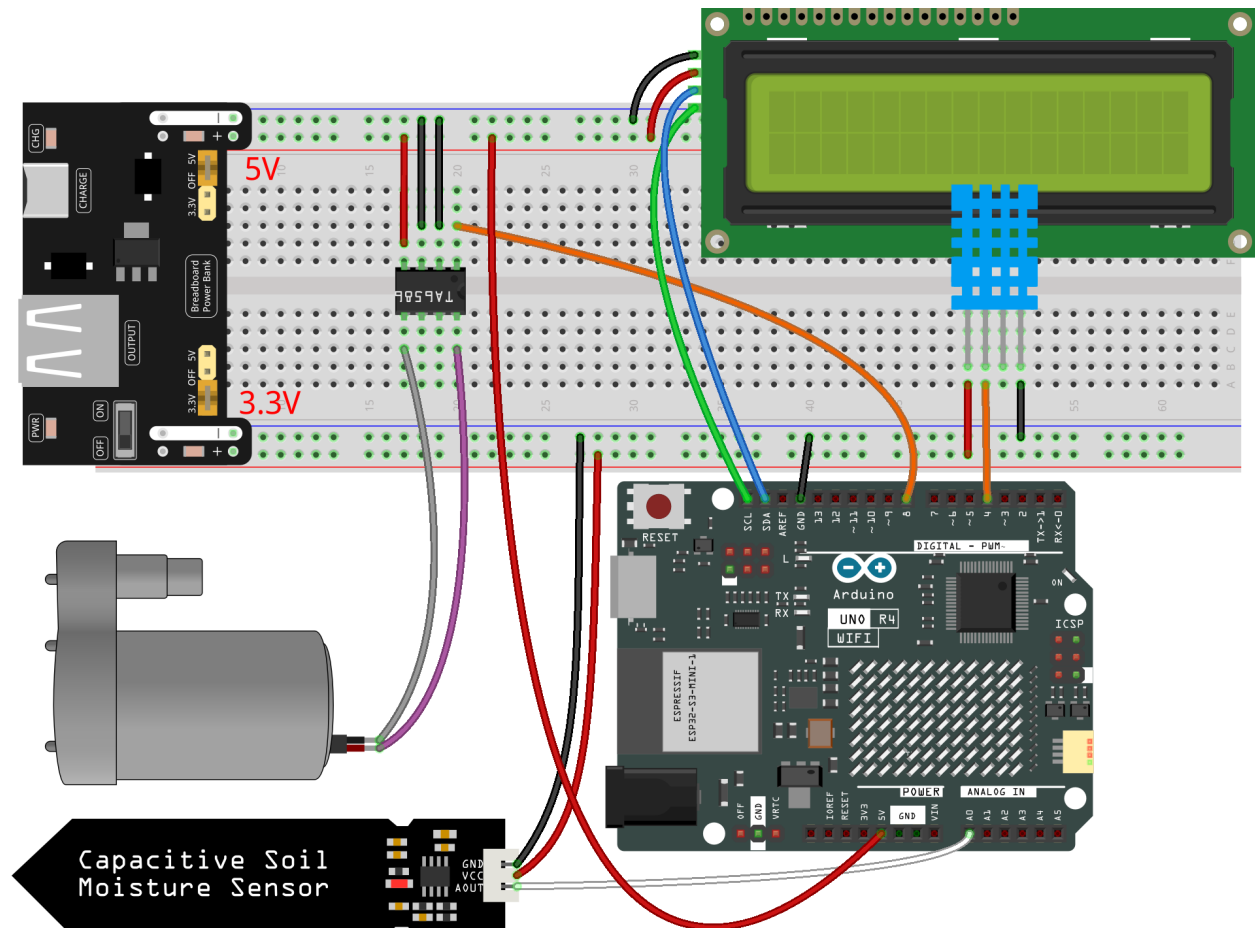
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

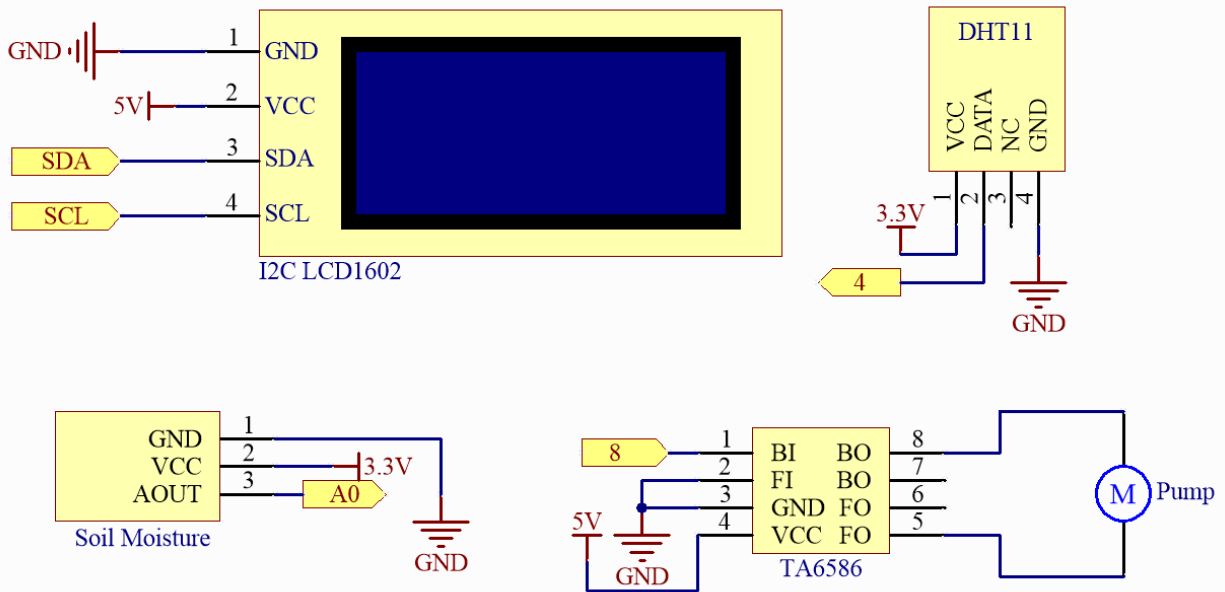
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>I2C LCD1602</i>	
<i>DC-Wasserpumpe</i>	-
<i>TA6586 - Motorsteuerungs-Chip</i>	-
<i>Bodenfeuchtigkeitsmodul</i>	
<i>Feuchtigkeitssensor-Modul</i>	
<i>Stromversorgungsmodul</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `08_plant_monitor.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\08_plant_monitor` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager und suchen Sie nach „DHT sensor library“ und „LiquidCrystal I2C“ und installieren Sie diese.

Wie funktioniert des?

Hier ist eine detaillierte Erklärung des Codes:

1. Bibliothekseinbindungen und Konstanten/Variablen:

Importieren Sie die Bibliotheken `Wire.h`, `LiquidCrystal_I2C.h` und `DHT.h`. Definieren Sie Pinnummern und andere Parameter, die mit dem DHT11, dem Bodenfeuchtigkeitssensor und der Wasserpumpe zusammenhängen.

2. `setup()`:

Initialisieren Sie die Pinmodi, die mit dem Bodenfeuchtigkeitssensor und der Wasserpumpe zusammenhängen. Schalten Sie die Wasserpumpe anfangs aus. Initialisieren Sie das LCD-Display und schalten Sie die Hintergrundbeleuchtung ein. Starten Sie den DHT-Sensor.

3. `loop()`:

Lesen Sie die Luftfeuchtigkeit und Temperatur vom DHT-Sensor. Lesen Sie die Bodenfeuchtigkeit vom Bodenfeuchtigkeitssensor. Zeigen Sie die Temperatur- und Luftfeuchtigkeitswerte auf dem LCD-Bildschirm an, löschen Sie dann den Bildschirm und zeigen Sie den Bodenfeuchtigkeitswert an. Entscheiden Sie, ob die Wasserpumpe aufgrund der Bodenfeuchtigkeit aktiviert werden soll. Wenn die Bodenfeuchtigkeit unter 500 (einem konfigurierbaren Schwellenwert) liegt, aktivieren Sie die Wasserpumpe für 1 Sekunde.

6.9 Zugangskontrollsystem

Die Hauptfunktion dieses Codes besteht darin, die Benutzerauthentifizierung mit einem RFID-Modul durchzuführen. Wenn die Authentifizierung erfolgreich ist, steuert er einen Schrittmotor, um die Tür zu öffnen, und gibt über einen Summer einen Ton aus, um das Ergebnis der Authentifizierung anzuzeigen. Scheitert die Authentifizierung, wird die Tür nicht geöffnet.

Sie können den seriellen Monitor öffnen, um die ID Ihrer RFID-Karte zu sehen und das Passwort in diesem Code neu zu konfigurieren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

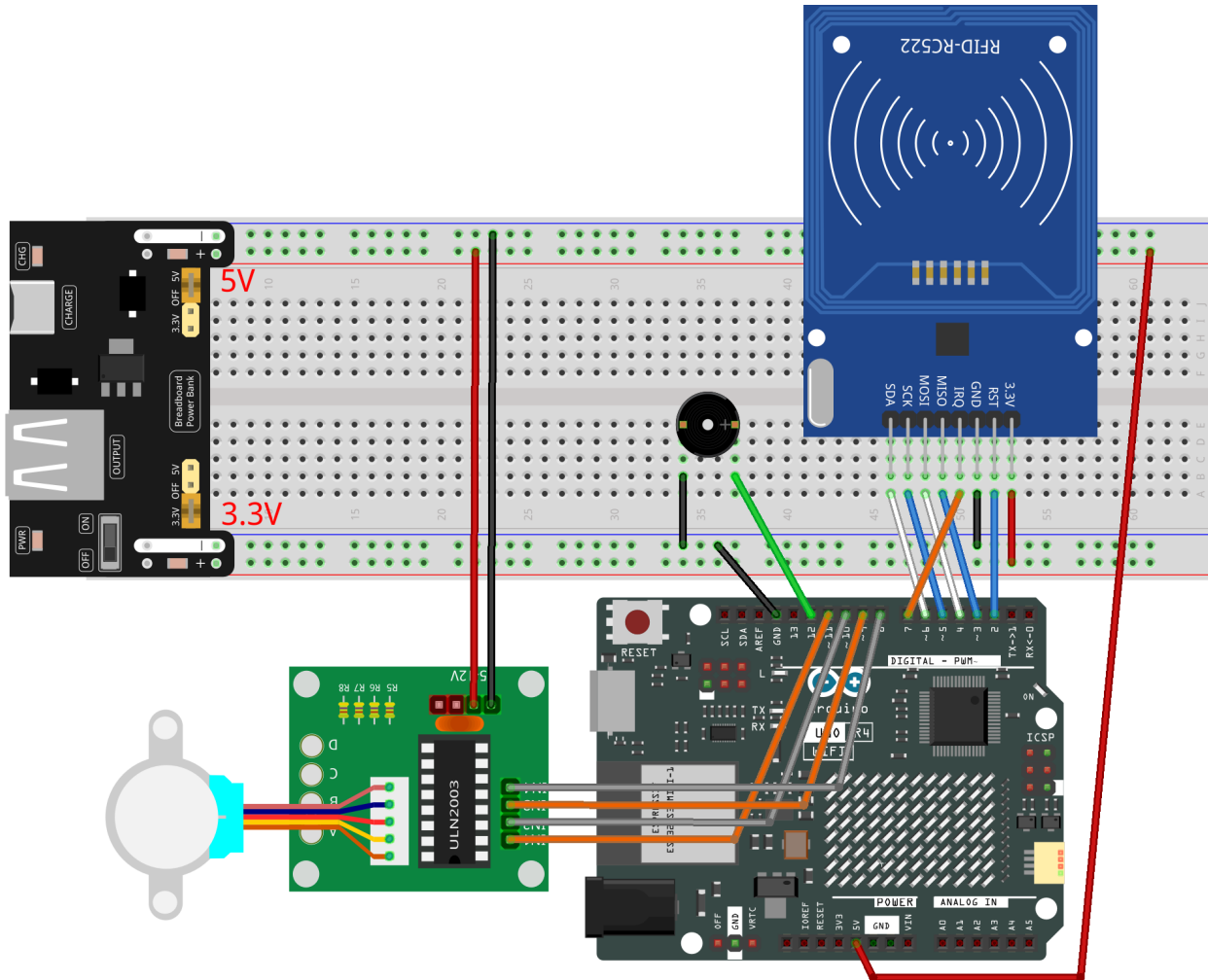
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

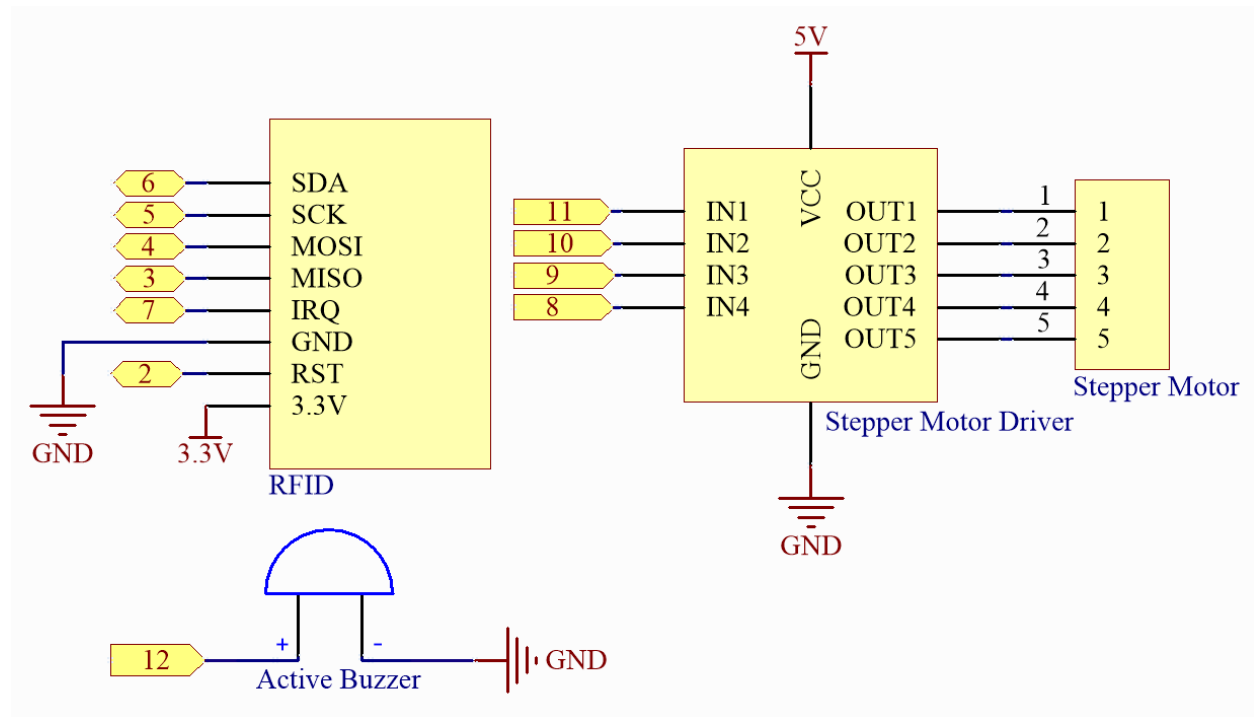
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>MFRC522-Modul</i>	
<i>Schrittmotor</i>	
<i>Summer</i>	-
<i>Stromversorgungsmodul</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `09_access_control_system.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project09_access_control_system` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung:

- Hier wird die RFID1-Bibliothek verwendet. Die Bibliothek finden Sie im Verzeichnis `elite-explorer-kit-main/library/`, oder Sie können hier [RFID1.zip](#) klicken, um sie herunterzuladen. Sehen Sie sich [Manuelle Installation](#) für eine Anleitung zur Installation an.

Wie funktioniert des?

Hier ist eine schrittweise Erklärung des Codes:

1. **Bibliotheksdateien einbinden:** Sie haben drei Bibliotheksdateien eingebunden: `rfid1.h`, `Stepper.h` und `Wire.h`. Diese Bibliotheken werden verwendet, um mit dem RFID-Modul, dem Schrittmotor und für die I2C-Kommunikation zu kommunizieren.
2. **Konstantendefinitionen:** Sie haben einige Konstanten definiert, darunter `ID_LEN` (Länge der ID), `stepsPerRevolution` (Schritte pro Umdrehung für den Schrittmotor), `rolePerMinute` (Geschwindigkeit des Schrittmotors), sowie die vier Schrittmotorpins (`IN1`, `IN2`, `IN3`, `IN4`), Buzzerpin (`buzPin`) und Variablen im Zusammenhang mit der Authentifizierung.
3. **Variablendefinitionen:** Sie haben Variablen wie ein Array definiert, um die gelesene Benutzer-ID (`userIdRead`), authentifizierte Benutzer-ID (`userId`) und eine boolesche Variable (`approved`) zur Anzeige einer erfolgreichen Authentifizierung zu speichern.

4. **Objektinstanzen:** Sie haben Instanzen von zwei Objekten erstellt: `RFID1 rfid` und `Stepper stepper` zur Interaktion mit dem RFID-Modul und Schrittmotor.
5. **setup():** In der Funktion `setup()` initialisieren Sie die Geschwindigkeit des Schrittmotors, stellen den Buzzerpin als Ausgang ein und initialisieren das RFID-Modul.
6. **loop():** In der Funktion `loop()` läuft Ihre Hauptlogik. Wenn `approved 0` ist (noch nicht authentifiziert), ruft es die Funktion `rfidRead()` auf, um Daten vom RFID-Modul zu lesen und löscht dann das Array `userIdRead`. Wenn `approved 1` ist (erfolgreiche Authentifizierung), ruft es die Funktion `openDoor()` auf, um die Tür zu öffnen, und setzt `approved` auf 0 zurück.
7. **beep():** Diese Funktion steuert den Summer basierend auf den bereitgestellten Parametern `duration` und `frequency`.
8. **verifyPrint():** Diese Funktion erzeugt unterschiedliche Summergeräusche basierend auf dem Parameter `result`, um anzuzeigen, ob die Authentifizierung erfolgreich war.
9. **openDoor():** Diese Funktion steuert den Schrittmotor, um die Tür bis zu einem bestimmten Winkel (`doorStep`) zu öffnen und wartet dann eine Weile, bevor die Tür geschlossen wird.
10. **rfidRead():** Diese Funktion liest Daten vom RFID-Modul, ruft zuerst `getId()` auf, um die Benutzer-ID zu erhalten, und dann `idVerify()`, um zu überprüfen, ob die Benutzer-ID mit der authentifizierten ID übereinstimmt.
11. **getId():** Diese Funktion holt die Benutzer-ID vom RFID-Modul und speichert sie im Array `userIdRead`. Sie gibt einen Piepton aus, wenn das Lesen fehlschlägt.
12. **idVerify():** Diese Funktion überprüft, ob die Benutzer-ID mit der authentifizierten ID übereinstimmt, und erzeugt einen Ton, der eine erfolgreiche oder fehlgeschlagene Authentifizierung anzeigt.

6.10 SPIEL - Zahl Erraten

„Zahl Erraten“ ist ein unterhaltsames Partyspiel, bei dem du und deine Freunde abwechselnd eine Zahl (0~99) eingeben. Der Bereich wird mit jeder Eingabe enger, bis ein Spieler die Antwort richtig errät. Der Spieler, der richtig rät, wird zum Verlierer erklärt und einer Strafe unterzogen. Zum Beispiel, wenn die geheime Zahl 51 ist, die die Spieler nicht sehen können, und Spieler 1 gibt 50 ein, ändert sich die Bereichsanzeige auf 50~99. Wenn Spieler 2 70 eingibt, wird der Bereich zu 50~70. Wenn Spieler 3 51 eingibt, ist er der Unglückliche. In diesem Spiel verwenden wir eine IR-Fernbedienung, um Zahlen einzugeben, und ein LCD, um Ergebnisse anzuzeigen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

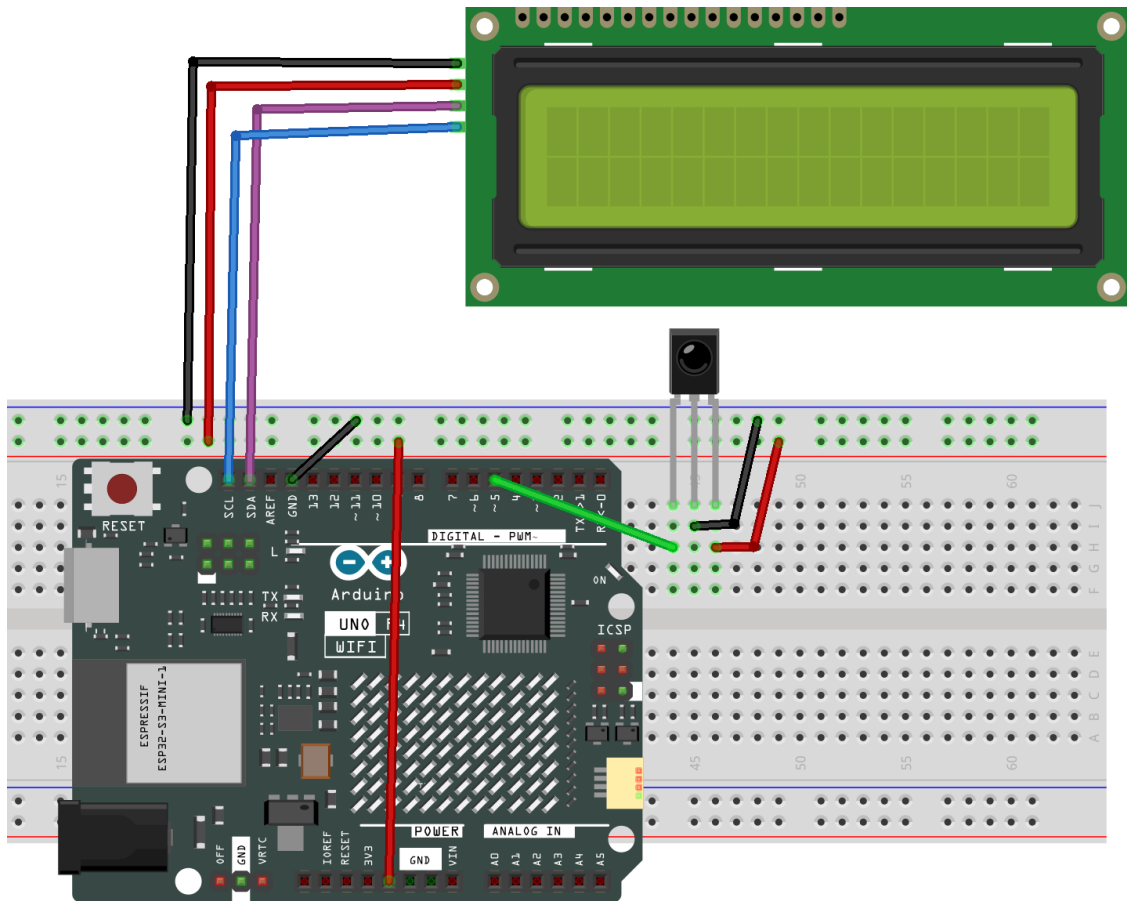
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

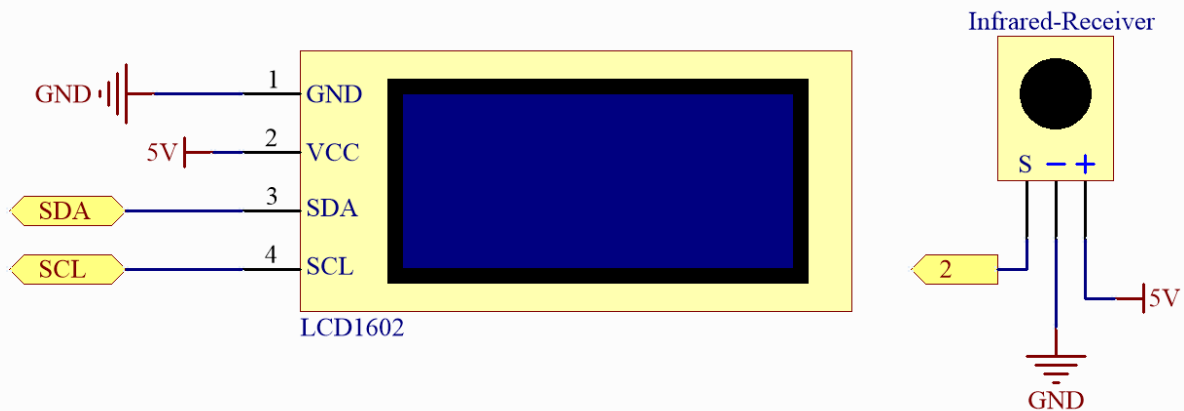
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>I2C LCD1602</i>	
<i>Infrarotempfänger</i>	

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `10_guess_number.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\10_guess_number` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager und suchen Sie nach „IRremote“ und „LiquidCrystal I2C“ und installieren Sie diese.

Wie funktioniert des?

1. Bibliotheksimporte und globale Variablendefinitionen:

Drei Bibliotheken werden importiert: `Wire` für die I2C-Kommunikation, `LiquidCrystal_I2C` zur Steuerung des LCD-Displays und `IRremote` zum Empfangen von Signalen der Infrarot-Fernbedienung. Mehrere globale Variablen werden definiert, um den Spielstand und die Einstellungen zu speichern.

2. `setup()`

Initialisieren Sie das LCD-Display und schalten Sie die Hintergrundbeleuchtung ein. Initialisieren Sie die serielle Kommunikation mit einer Baudrate von 9600. Starten Sie den Infrarotempfänger. Rufen Sie die Funktion `initNewValue()` auf, um den anfänglichen Spielzustand einzustellen.

3. `loop()`

Überprüfen Sie, ob ein Signal von der Infrarot-Fernbedienung empfangen wurde. Dekodieren Sie das empfangene Infrarotsignal. Aktualisieren Sie den Spielstand oder führen Sie entsprechende Aktionen basierend auf dem dekodierten Wert (Zahl oder Befehl) aus.

4. `initNewValue()`

Verwenden Sie `analogRead`, um den Zufallszahlensamen zu initialisieren und zu gewährleisten, dass jedes Mal unterschiedliche Zufallszahlen generiert werden. Generieren Sie eine Zufallszahl zwischen 0 und 98 als Glückszahl (die Zahl, die die Spieler erraten müssen). Setzen Sie die Hinweise für die obere und untere Grenze zurück. Zeigen Sie eine Begrüßungsnachricht auf dem LCD an. Setzen Sie die Eingabenummer zurück.

5. `detectPoint()`

Überprüfen Sie das Verhältnis zwischen der Eingabenummer des Spielers und der Glückszahl. Wenn die Eingabenummer größer als die Glückszahl ist, aktualisieren Sie den Hinweis für die obere Grenze. Wenn die Eingabenummer kleiner als die Glückszahl ist, aktualisieren Sie den Hinweis für die untere Grenze.

benummer kleiner als die Glückszahl ist, aktualisieren Sie den Hinweis für die untere Grenze. Wenn der Spieler die richtige Zahl eingibt, setzen Sie die Eingabe zurück und geben Sie wahr zurück.

6. lcdShowInput()

Zeigen Sie die Eingabe des Spielers und die aktuellen Hinweise für die obere und untere Grenze auf dem LCD an. Wenn der Spieler richtig rät, zeigen Sie eine Erfolgsmeldung an und pausieren Sie für 5 Sekunden, bevor Sie das Spiel neu starten.

6.11 SPIEL - Flucht

Dieses Spiel heißt „Flucht“. Das Ziel des Spielers ist es, den MPU6050-Sensor zu neigen, um einen Pixel auf der LED-Matrix zu bewegen und zu versuchen, ihn durch eine Öffnung im Rand der Matrix (den Ausgang) zu manövrieren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

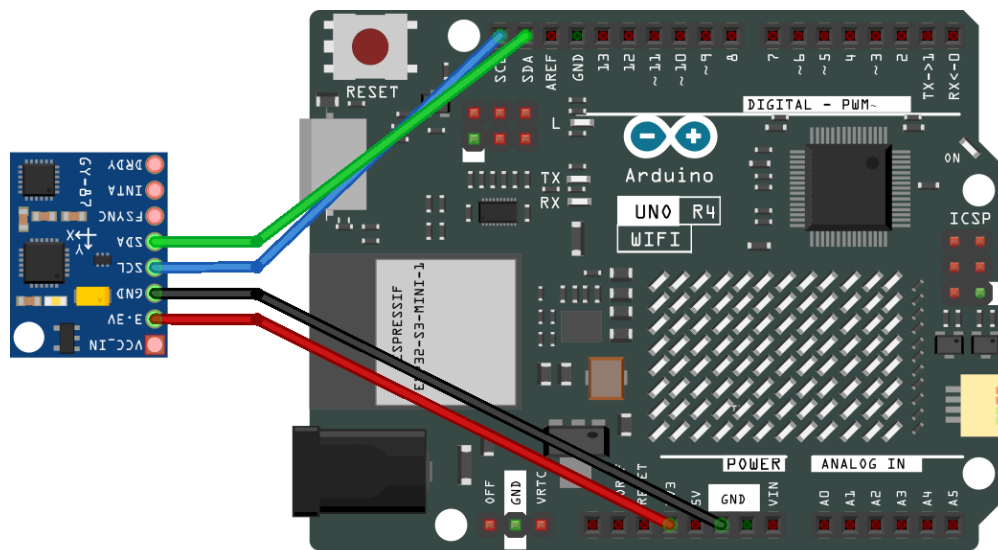
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

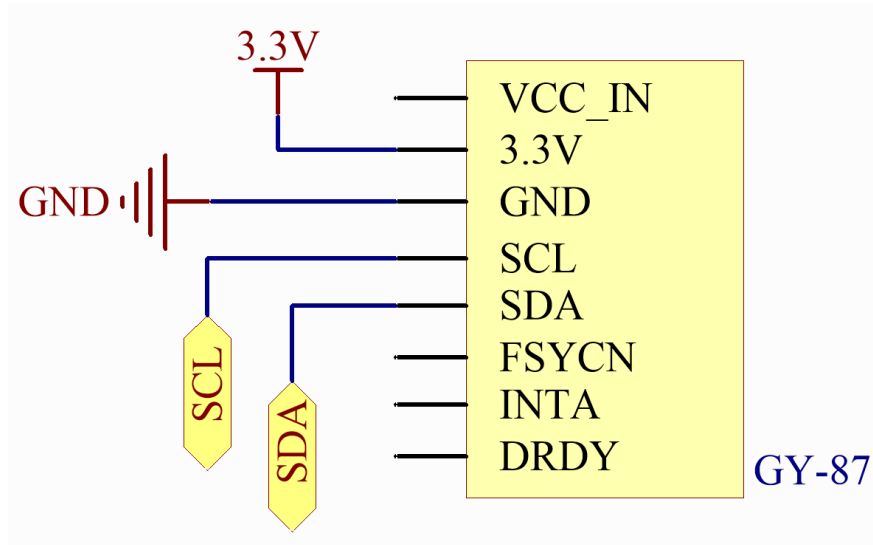
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>GY-87 IMU-Modul</i>	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `11_escape_square.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\11_escape_square` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager und suchen Sie nach „Adafruit MPU6050“ und installieren Sie diese.

Wie funktioniert des?

Hier ist eine detaillierte Erklärung des Codes:

1. Bibliotheksimporte und globale Variablen:

Importierte Bibliotheken umfassen die LED-Matrix, Wire (für I2C-Kommunikation), MPU6050 (Bewegungssensor) und die Adafruit_Sensor-Bibliothek. Initialisierung von MPU6050 und LED-Matrix-Objekten. Definition globaler Variablen wie `pixelX` und `pixelY` (Pixelposition), `gapStart` und `side` (Startposition der Lücke und welche Seite), `level` (Schwierigkeitsgrad des Spiels) und mehr.

2. `setup()`:

Initialisieren Sie die LED-Matrix und zeichnen Sie die Matrix mit der Lücke. Initialisieren Sie die serielle Kommunikation und prüfen Sie, ob der MPU6050-Sensor korrekt startet, und stellen Sie seinen Beschleunigungsbereich auf 2g ein.

3. `loop()`:

Aktualisieren Sie regelmäßig die Position des Pixels basierend auf den MPU6050-Sensorwerten. Verschieben Sie regelmäßig die Position der Lücke. Laden Sie das neue Pixel-Layout und rendern Sie es auf der LED-Matrix. Überprüfen Sie, ob das Pixel durch die Lücke gegangen ist. Wenn ja, warten Sie 1,5 Sekunden, um den Erfolg anzuzeigen, erhöhen Sie den Schwierigkeitsgrad des Spiels und setzen Sie die Pixelposition zurück.

4. Weitere Funktionen:

- `drawSquareWithGap()`: Zeichnen Sie einen 8x8-Rand und erstellen Sie eine Lücke darin.

- `createGap()`: Erstellen Sie eine 2 Pixel lange Lücke auf der angegebenen Seite.
- `moveGap()`: Verschieben Sie die Position der Lücke basierend auf der aktuellen Seite und `gapStart` und ändern Sie bei Bedarf die Seite.
- `movePixelBasedOnMPU()`: Lesen Sie die Beschleunigungsdaten von MPU6050. Bewegen Sie die Pixelposition basierend auf den Beschleunigungsdaten (zurücksetzen, wenn das Pixel außerhalb der Grenzen oder in Wände geht).
- `resetPixel()`: Setzen Sie die Pixelposition auf die Mitte der Matrix zurück.
- `checkPixelPosition()`: Überprüfen Sie, ob sich das Pixel auf der Lücke befindet. Wenn ja, erhöhen Sie den Schwierigkeitsgrad des Spiels und setzen Sie das Pass-Flag auf wahr.

6.12 SPIEL - Pong

Dies ist ein einfaches Pong-Spiel, das mit einem OLED-Display und einem Arduino-Board entwickelt wurde. Im Pong-Spiel treten Spieler gegen den Computer an, indem sie einen vertikalen Schläger steuern, um einen hüpfenden Ball zurückzuprallen. Das Ziel ist es, zu verhindern, dass der Ball an der Kante Ihres Schlägers vorbeigeht, sonst erzielt der Gegner Punkte.

Die Spielmechanik kann in folgende Teile unterteilt werden:

1. Ballbewegung - Der Ball bewegt sich in seiner aktuellen Richtung mit einer festgelegten Geschwindigkeit. Jedes Mal, wenn der Ball mit einem Schläger kollidiert, nimmt seine Geschwindigkeit zu, was das Spiel herausfordernder macht.
2. Schlägerbewegung - Wird verwendet, um die Bewegung des Balls zu blockieren, der Schläger kann sich nach oben oder unten bewegen. Spieler steuern ihren eigenen Schläger mit Knöpfen, während der Schläger des Computers automatisch der Position des Balls folgt.
3. Punktzahl - Immer wenn der Ball über den linken oder rechten Rand des Bildschirms hinausgeht, erzielt der entsprechende Spieler oder CPU Punkte.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

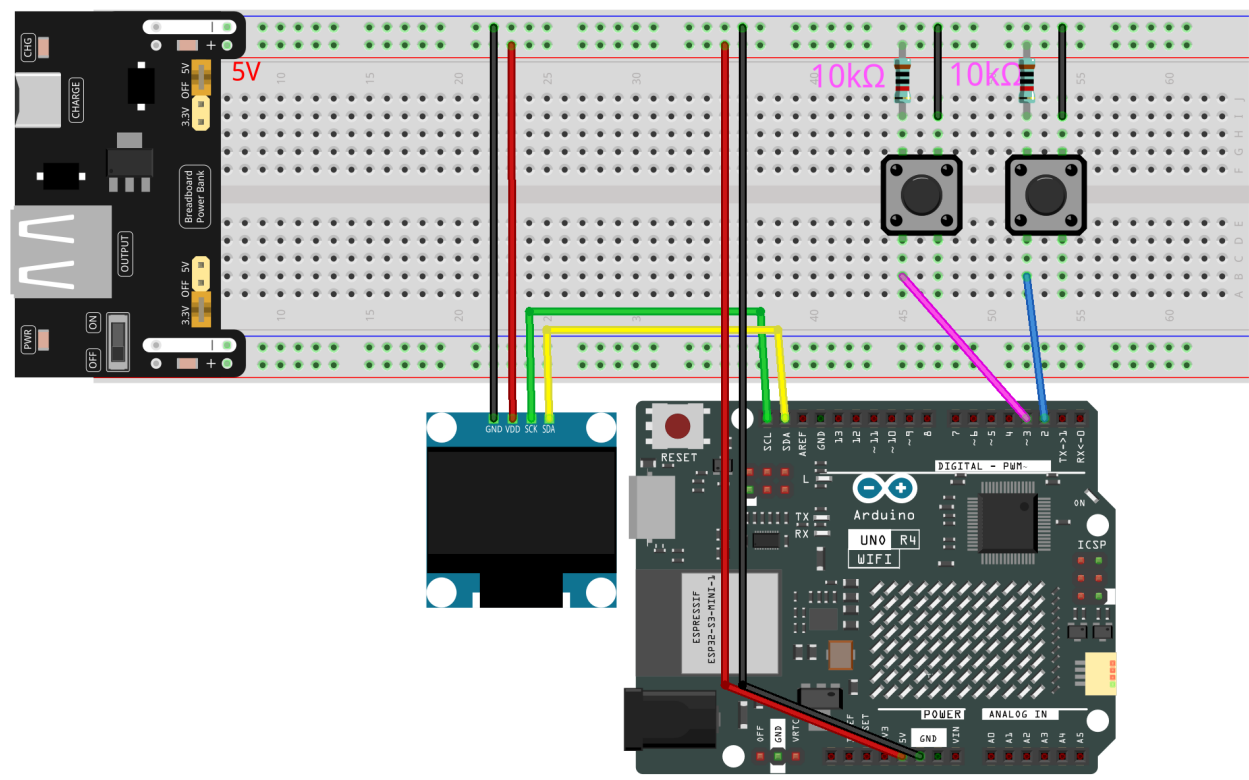
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

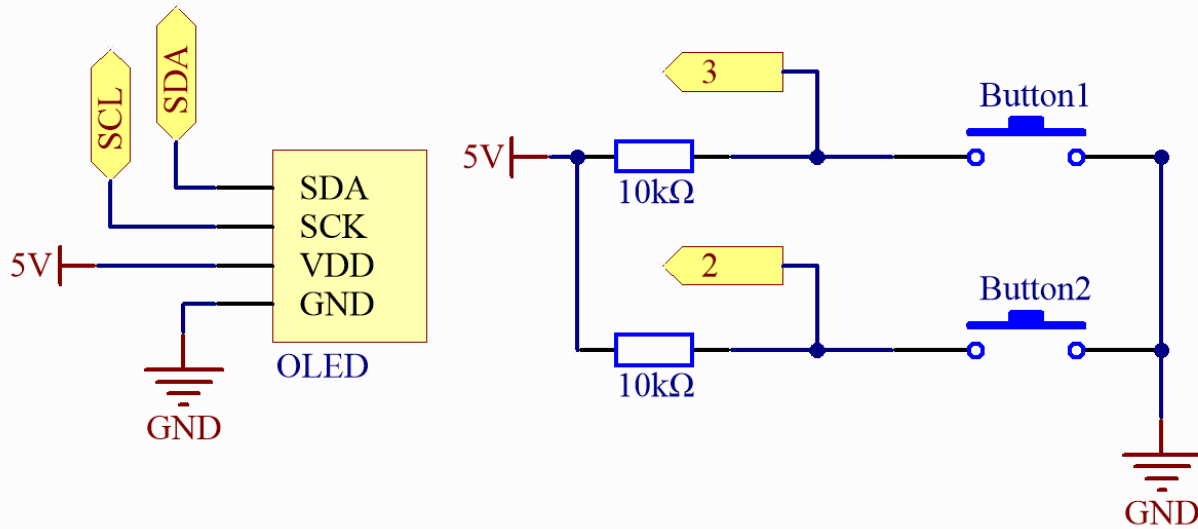
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Steckbrett	
Jumperkabel	
Widerstand	
OLED-Display-Modul	
Knopf	
Stromversorgungsmodul	-

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `12_pong_oled.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\12_pong_oled` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager und suchen Sie nach „Adafruit SSD1306“ und „Adafruit GFX“ und installieren Sie diese.

Wie funktioniert des?

Die Programmstruktur kann in die folgenden fünf Teile unterteilt werden:

1. Notwendige Bibliotheken importieren - Verwendet zur Steuerung des OLED-Bildschirms und zum Lesen von Tasteneingaben.
2. Konstanten und globale Variablen definieren:
Definitionen für OLED-Bildschirmbreite und -höhe. Definitionen für Tasten und OLED-Reset-Pins. Position, Geschwindigkeit, Größe und Richtung des Balls und der Schläger. Punktzahlen für Spieler und CPU.
3. Initialisierung:
Initialisieren Sie die serielle Kommunikation, den OLED-Bildschirm und zeigen Sie die anfängliche Oberfläche an. Stellen Sie Tasten als Eingänge ein und verbinden Sie Pull-up-Widerstände. Zeichnen Sie das Spielfeld.
4. Hauptschleife:
Lesen Sie die Zustände der Tasten. Bewegen Sie den Ball basierend auf der eingestellten Aktualisierungsrate. Erkennen Sie Kollisionen zwischen dem Ball und den Schlägern oder Wänden und passen Sie die Richtung und Geschwindigkeit des Balls entsprechend an. Aktualisieren Sie den Bildschirm mit Punktzahlen basierend auf Punktereignissen. Aktualisieren Sie die Positionen der Schläger.
5. Zusätzliche Funktionen:

- crossesPlayerPaddle und crossesCpuPaddle - Wird verwendet, um zu erkennen, ob der Ball mit dem Schläger des Spielers oder der CPU kollidiert.
- drawCourt - Zeichnet das Spielfeld auf dem OLED-Bildschirm.
- displayScore - Zeigt die Punktzahlen des Spielers und der CPU auf dem Bildschirm an.

6.13 SPIEL - Snake

Dieses Beispiel implementiert das klassische Snake-Spiel auf einer 8x12 LED-Matrix mit dem R4 Wifi Board. Spieler steuern die Richtung der Schlange mit einem Zwei-Achsen-Joystick.

Benötigte Komponenten

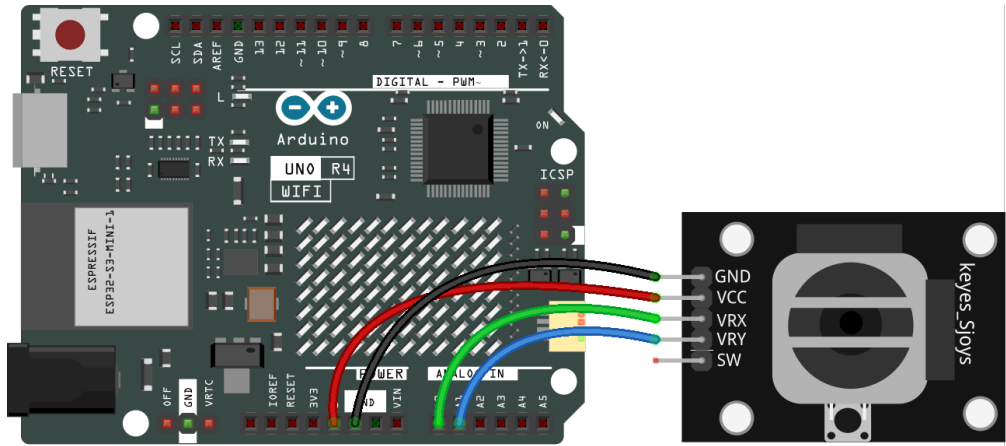
Für dieses Projekt benötigen wir die folgenden Komponenten.
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

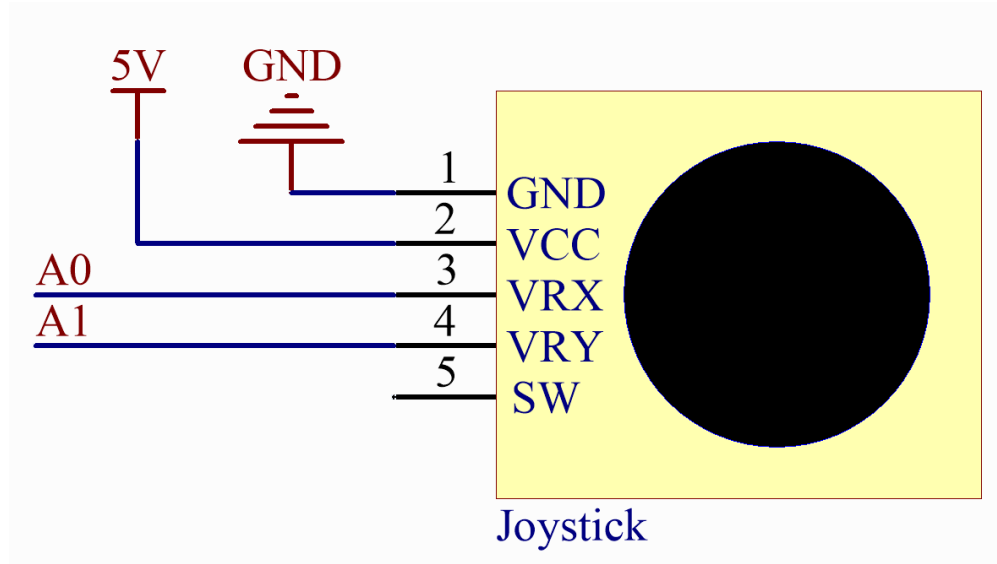
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Jumperkabel	
Joystick-Modul	

Verdrahtung



Schaltplan



Code

Bemerkung:

- Sie können die Datei `13_snake.ino` direkt unter dem Pfad `elite-explorer-kit-main\fun_project\13_snake` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Wie funktioniert des?

Hier ist eine detaillierte Erklärung des Codes:

1. Variablendefinition und -initialisierung

Importieren Sie die Bibliothek `Arduino_LED_Matrix` für LED-Matrix-Operationen. `matrix` ist eine Instanz der LED-Matrix. `frame` und `flatFrame` sind Arrays, die verwendet werden, um Pixelinformationen auf dem Bildschirm zu speichern und zu verarbeiten. Die Schlange wird als ein Array von `Point`-Strukturen dargestellt, wobei jeder Punkt eine x- und y-Koordinate hat. `food` repräsentiert die Position des Futters. `direction` ist die aktuelle Bewegungsrichtung der Schlange.

2. `setup()`

Initialisieren Sie die X- und Y-Achsen des Joysticks als Eingänge. Starten Sie die LED-Matrix. Initialisieren Sie die Startposition der Schlange in der Mitte des Bildschirms. Generieren Sie die anfängliche Position des Futters zufällig.

3. `loop()`

Bestimmen Sie die Richtung der Schlange anhand der Ablesungen vom Joystick. Bewegen Sie die Schlange. Überprüfen Sie, ob der Kopf der Schlange mit dem Futter kollidiert. Wenn ja, wächst die Schlange und neues Futter wird an einem neuen Ort generiert. Überprüfen Sie, ob die Schlange mit sich selbst kollidiert. Wenn ja, setzen Sie das Spiel zurück. Zeichnen Sie den aktuellen Spielstand (Positionen von Schlange und Futter) auf der LED-Matrix. Fügen Sie eine Verzögerung hinzu, um die Geschwindigkeit des Spiels zu steuern.

4. `moveSnake()`

Bewegen Sie jeden Teil der Schlange an die Position des vorherigen Teils, beginnend am Schwanz und bewegend zum Kopf. Bewegen Sie den Kopf der Schlange basierend auf ihrer Richtung.

5. generateFood()

Generieren Sie alle möglichen Futterpositionen. Überprüfen Sie, ob jede Position mit irgendeinem Teil der Schlange überlappt. Wenn es nicht überlappt, wird die Position als möglicher Futterort betrachtet. Wählen Sie zufällig einen möglichen Futterort aus.

6. drawFrame()

Löschen Sie das aktuelle Frame. Zeichnen Sie die Schlange und das Futter auf dem Frame. Flachen Sie das zweidimensionale Frame-Array in ein eindimensionales Array (flatFrame) ab und laden Sie es auf die LED-Matrix.

Das Elite Explorer Kit nutzt das eingebaute ESP32-S3 WLAN- und Bluetooth-Modul auf dem Arduino UNO R4 WiFi-Board, wodurch eine Vielzahl von einfachen und unterhaltsamen IoT-Projekten ermöglicht wird. Die WLAN-Verbindung erlaubt es Ihnen, Ihren Arduino mit dem Internet und Cloud-Plattformen für IoT-Experimente zu verbinden. Und Bluetooth bietet kurze drahtlose Kommunikationsmöglichkeiten.

Mit WLAN können Sie Projekte wie einen einfachen Webserver zum Fernsteuern einer LED erstellen, mit der Arduino IoT Cloud interagieren, um Sensoren zu überwachen, Sicherheitswarnungen mit IFTTT unter Verwendung eines PIR-Sensors erstellen und ein cloudbasiertes Rufsystem mit MQTT realisieren. Wir bieten Schritt-für-Schritt-Anleitungen zur Umsetzung dieser und anderer vernetzter IoT-Projekte.

Die Bluetooth-Funktionalität ermöglicht lokale drahtlose Projekte wie den Austausch von Nachrichten auf einem LCD-Bildschirm, die Überwachung von Temperatur- und Feuchtigkeitsdaten von Sensoren auf einer Smartphone-App.

7.1 Einfacher Webserver

Dieses einfache Arduino-Programm dient dazu, einen grundlegenden WiFi-Webserver zu erstellen. Benutzer können damit den Ein- und Ausschaltzustand einer LED auf dem Arduino-Board über einen Webbrowser steuern.

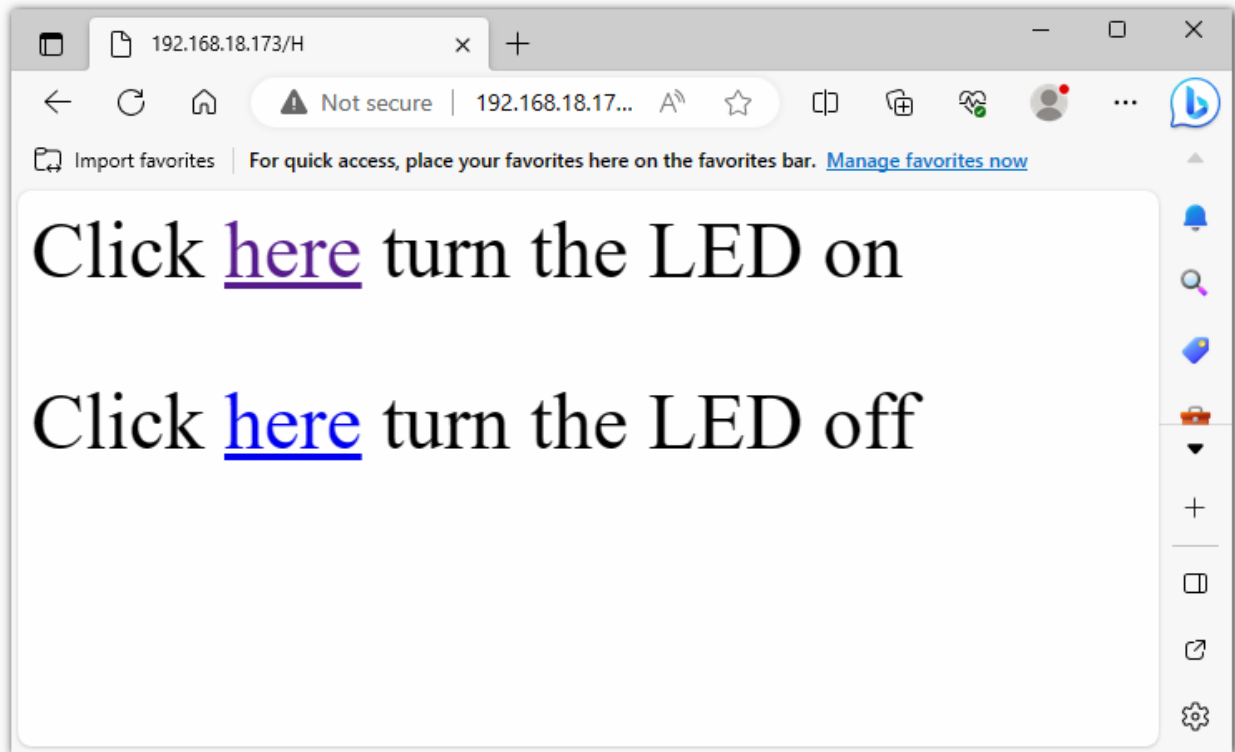
Ausführen des Codes

Bemerkung:

- Die Datei `01_simple_webserver.ino` können Sie direkt unter dem Pfad `elite-explorer-kit-main\iot_project\01_simple_webserver` öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Bemerkung: Im Code werden SSID und Passwort in `arduino_secrets.h` gespeichert. Bevor Sie dieses Beispiel hochladen, müssen Sie diese mit Ihren eigenen WiFi-Anmeldedaten ändern. Zusätzlich sollten Sie diese Informationen geheim halten, wenn Sie den Code teilen oder speichern, um Sicherheitsgründe zu gewährleisten.

Nach dem Hochladen des Codes können Sie die IP-Adresse im seriellen Monitor sehen. Geben Sie diese IP-Adresse in Ihren Webbrowser ein, um die LED an Bord ein-/auszuschalten.



Wie funktioniert des?

Hier ist eine Erklärung des Codes:

1. Header-Dateien und globale Variablen:

- `#include "WiFiS3.h"`: Diese beinhaltet die WiFi-Bibliothek für das Verbinden und Verwalten von WiFi. Diese Bibliothek ist im Arduino UNO R4 Core enthalten, eine zusätzliche Installation ist nicht erforderlich.
- `#include "arduino_secrets.h"`: Diese beinhaltet sensible WiFi-Verbindungsdaten wie SSID und Passwort.
- `ssid, pass, keyIndex`: Dies sind die Netzwerkanmeldeinformationen für die WiFi-Verbindung.
- `led, status, server`: Diese definieren den LED-Pin, den WiFi-Status und das Webserver-Objekt.

2. `setup()`:

- Beginnen Sie mit der seriellen Kommunikation.
- Überprüfen Sie das Vorhandensein des WiFi-Moduls.
- Überprüfen Sie, ob die Firmware-Version des WiFi-Moduls aktuell ist.
- Versuchen Sie, sich mit dem WiFi-Netzwerk zu verbinden.
- Starten Sie den Webserver.
- Drucken Sie den WiFi-Status aus.

3. `loop()`:

- Überprüfen Sie auf neue Webclient-Verbindungen.

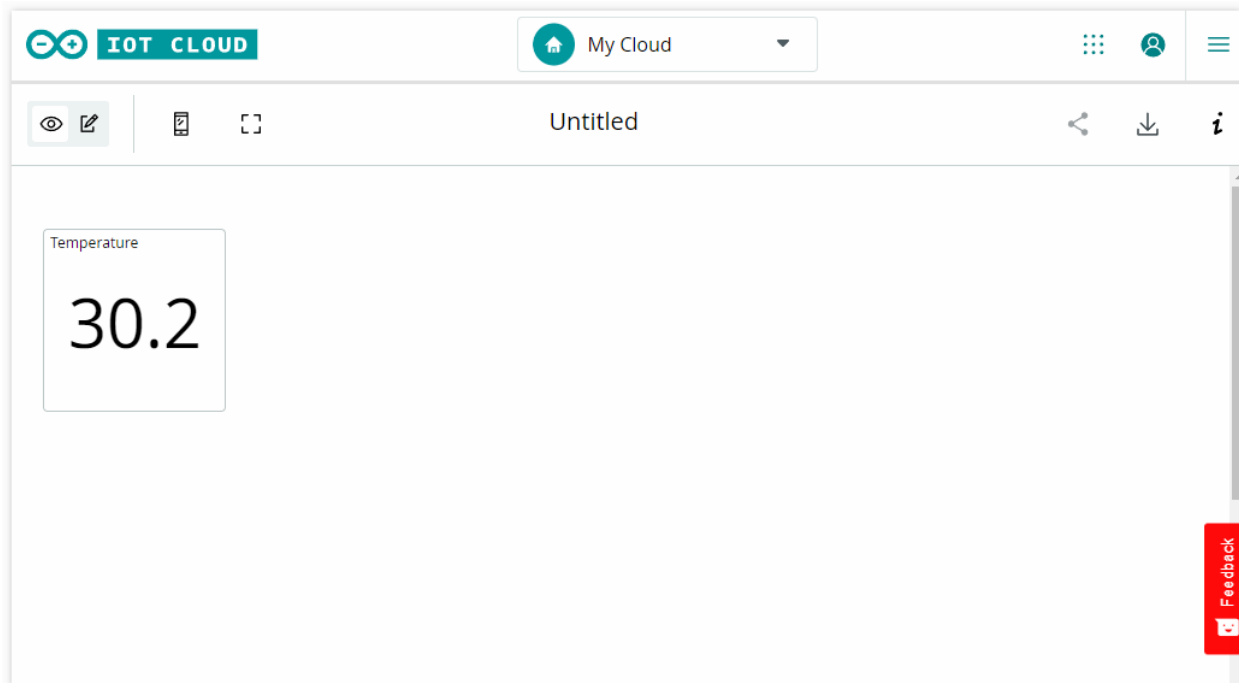
- Wenn es Client-Verbindungen gibt, lesen Sie deren eingehende HTTP-Anfragen.
- Basierend auf den Anfragen können Sie den Ein-/Ausschaltzustand der LED steuern. Wenn beispielsweise die Anfrage „GET /H“ lautet, wird die LED eingeschaltet; bei „GET /L“ wird sie ausgeschaltet.
- Senden Sie eine HTTP-Antwort, um den Benutzer anzuleiten, wie er die LED steuern kann.
- Trennen Sie die Verbindung zum Client.

4. `printWifiStatus()`:

- Drucken Sie die verbundene WiFi-SSID aus.
- Drucken Sie die IP-Adresse des Arduino-Boards aus.
- Drucken Sie die empfangene Signalstärke aus.
- Erklären Sie, wie man diese Seite in einem Webbrowser ansehen kann.

7.2 Arduino IoT Cloud

Dieses Beispiel zeigt einen Code zur Kommunikation mit der Arduino IoT Cloud. Ziel ist es, eine Verbindung zur Arduino IoT Cloud herzustellen und mit Cloud-Variablen zu interagieren. Hier senden wir die vom DHT11-Sensor gelesenen Temperaturwerte an die Arduino IoT Cloud, was uns ermöglicht, diese aus der Cloud zu überwachen.



Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

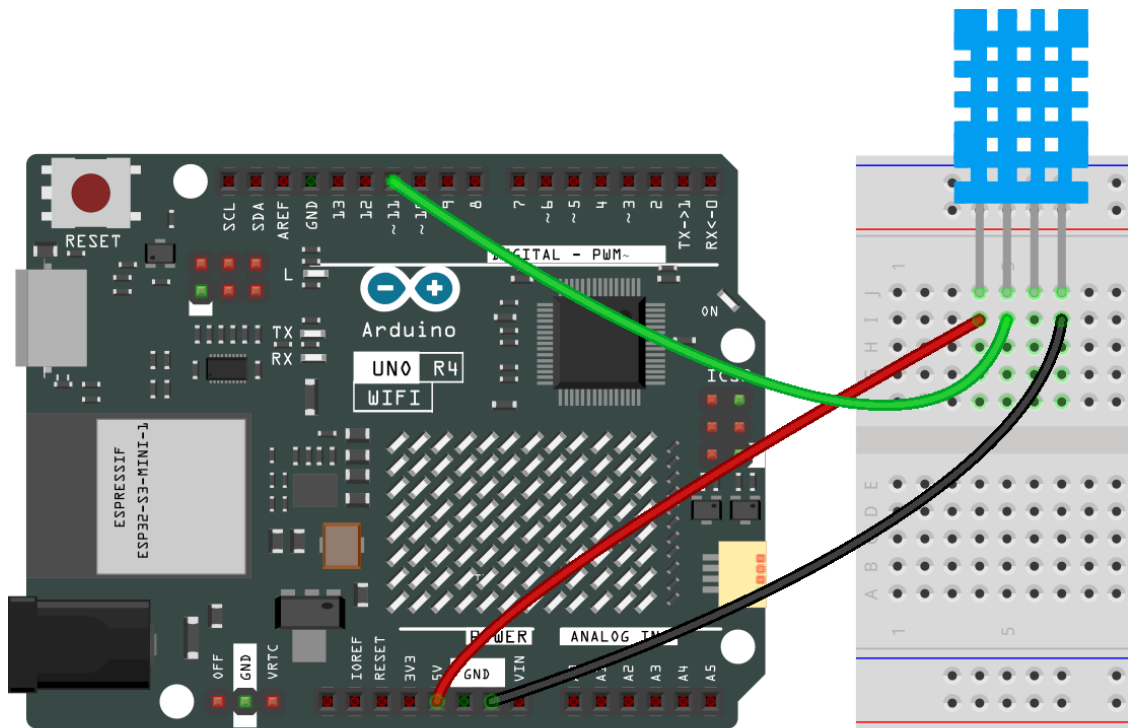
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

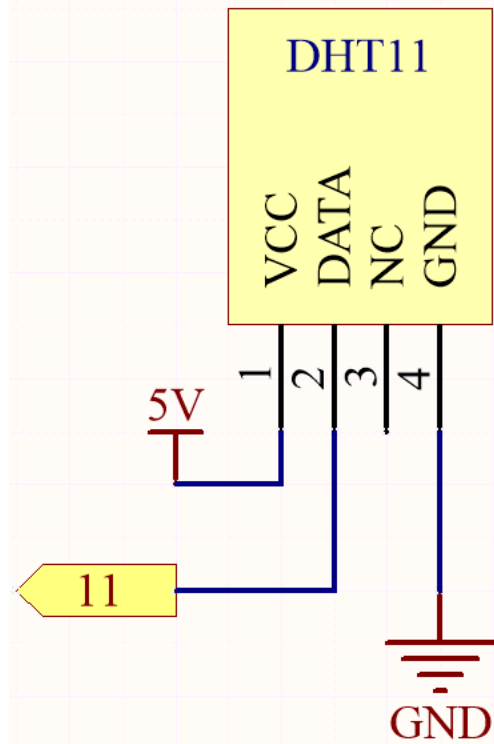
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Feuchtigkeitssensor-Modul</i>	

Verdrahtung

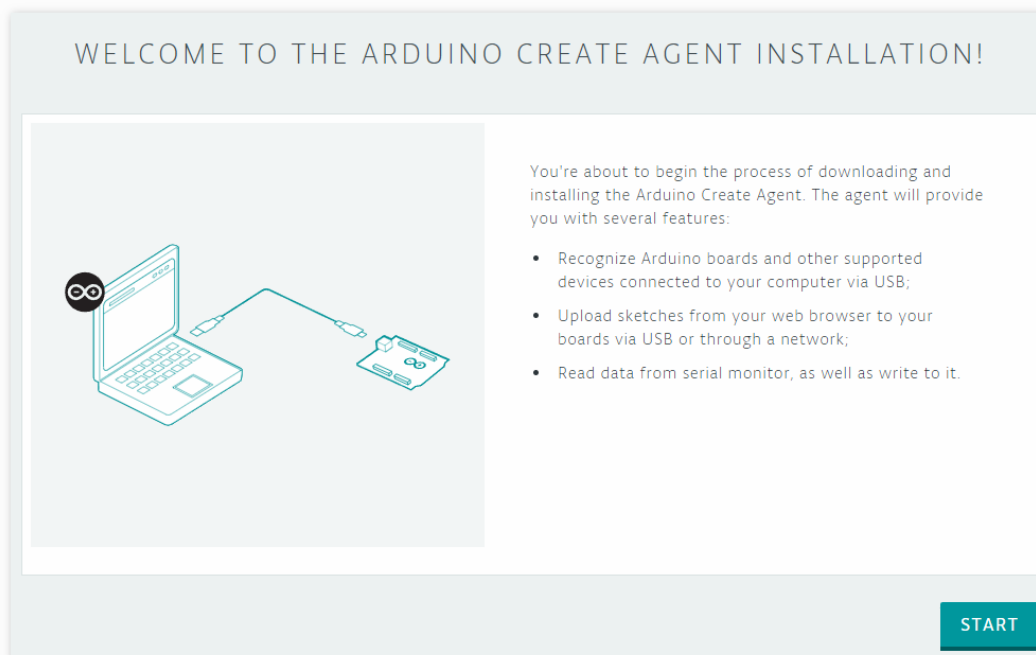


Schaltplan

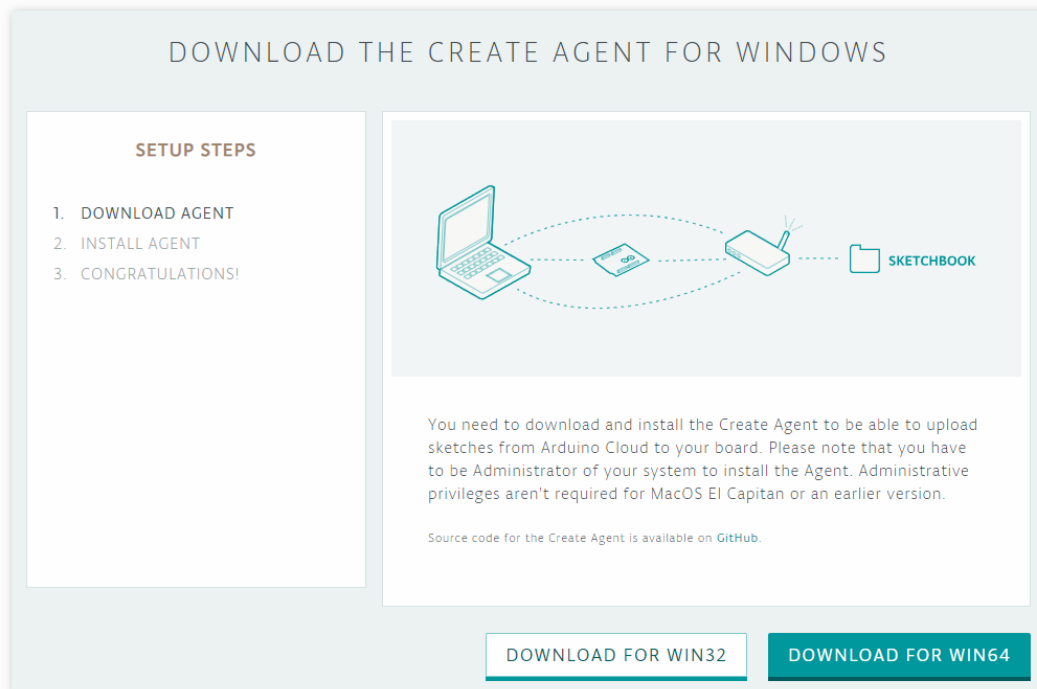


Installation des Arduino Create Agent

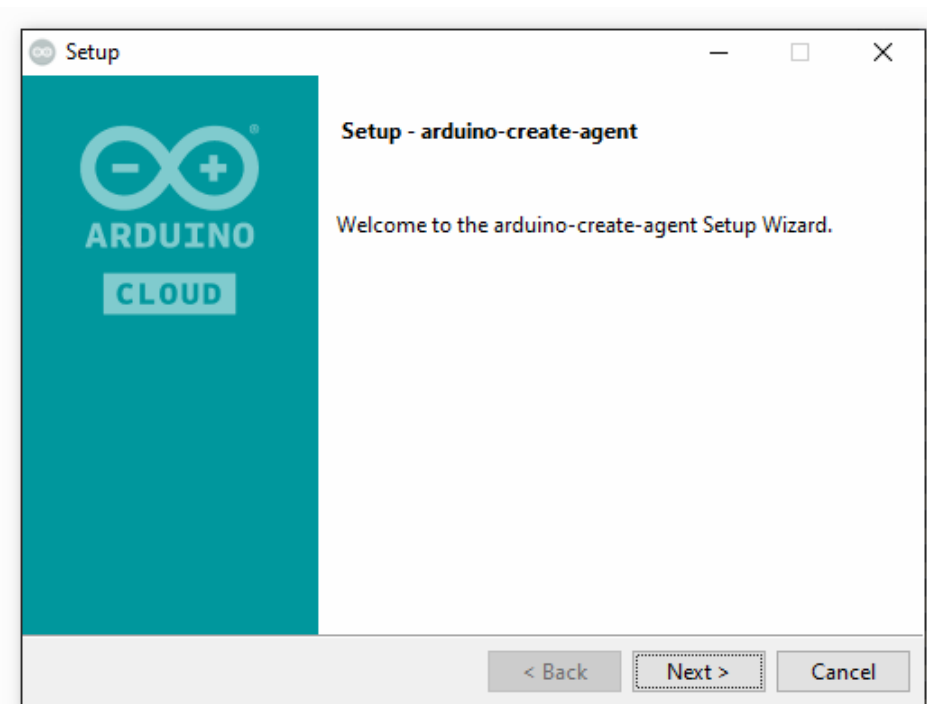
1. Besuchen Sie die Adresse <https://create.arduino.cc/getting-started/plugin/welcome>.
2. Klicken Sie auf START.



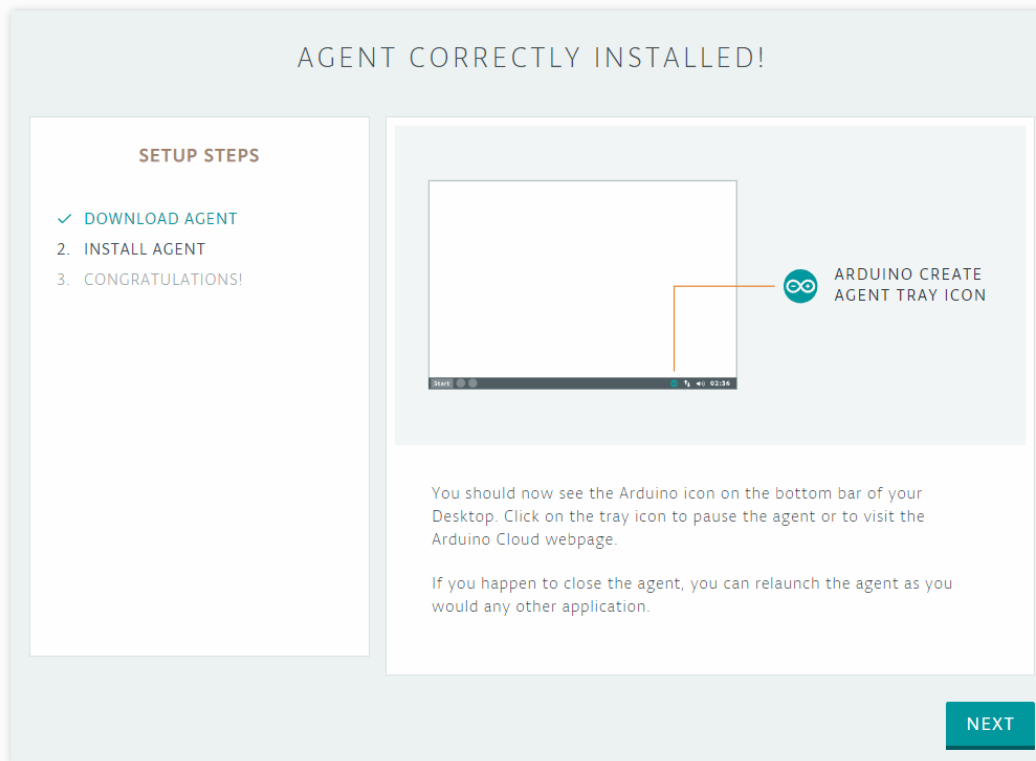
3. Wählen Sie die Version, die zu Ihrem Computer passt, und es wird ein Installationspaket heruntergeladen.



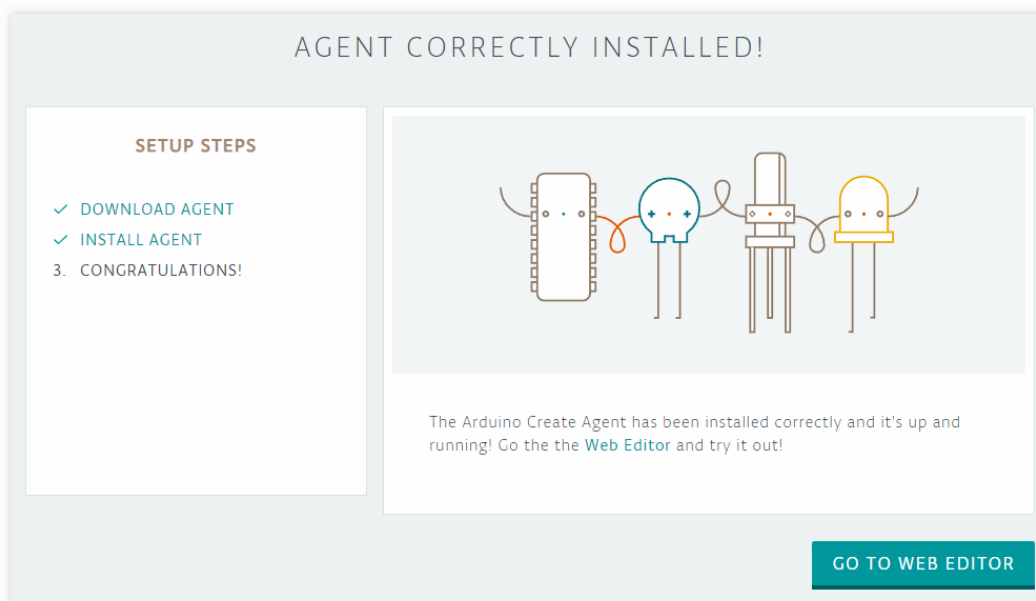
4. Installieren Sie den Agent.



5. Nach der Installation gehen Sie zurück zu Ihrem Browser, und Sie sehen die folgende Oberfläche.

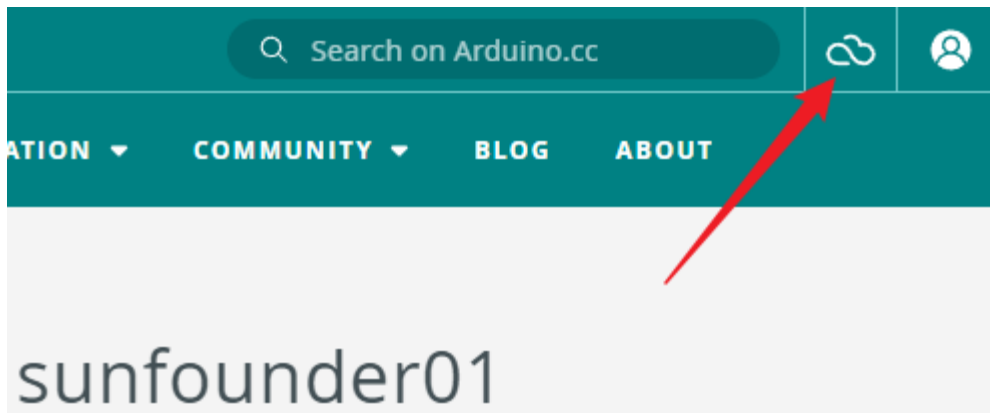


6. Klicken Sie auf WEITER, und dann können Sie ZUM Web-Editor GEHEN.

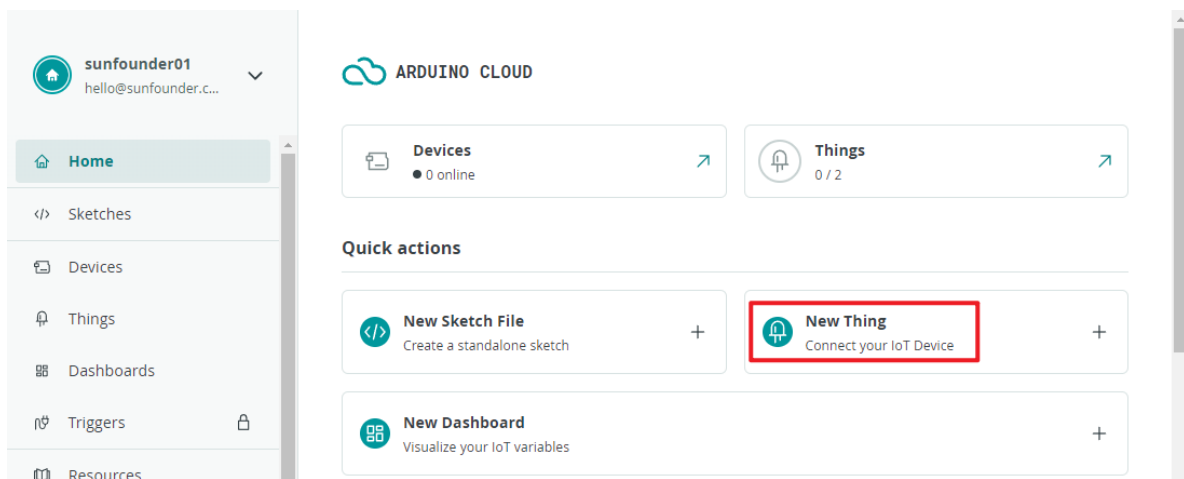


Verwendung der Arduino IoT Cloud

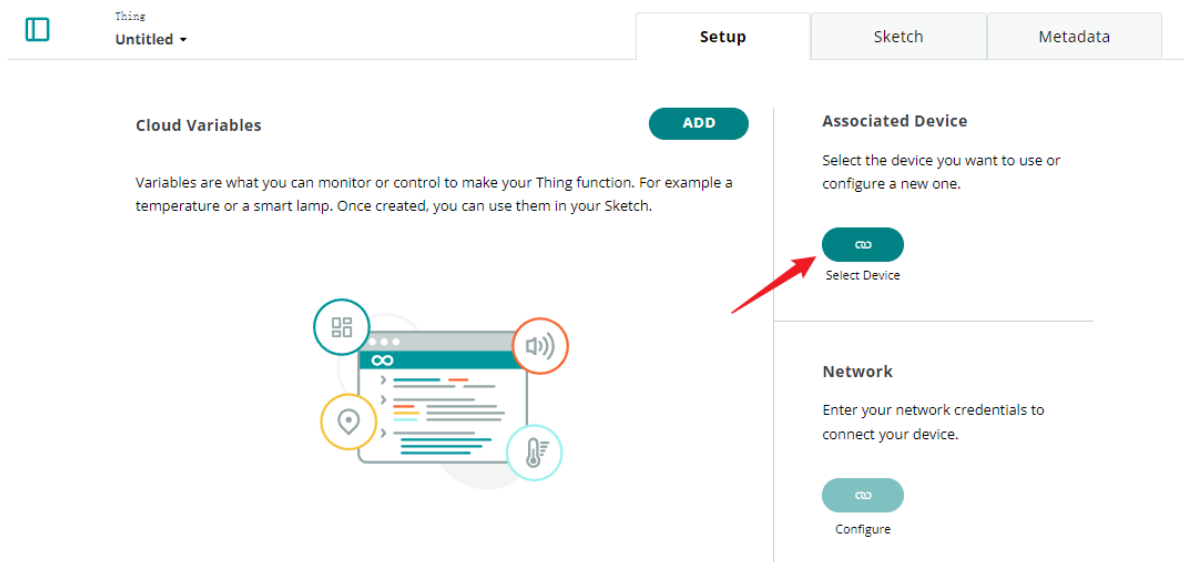
1. Zuerst müssen Sie sich bei Arduino anmelden oder registrieren.
<https://login.arduino.cc/login>
2. Nach dem Einloggen klicken Sie oben rechts auf IoT Cloud.



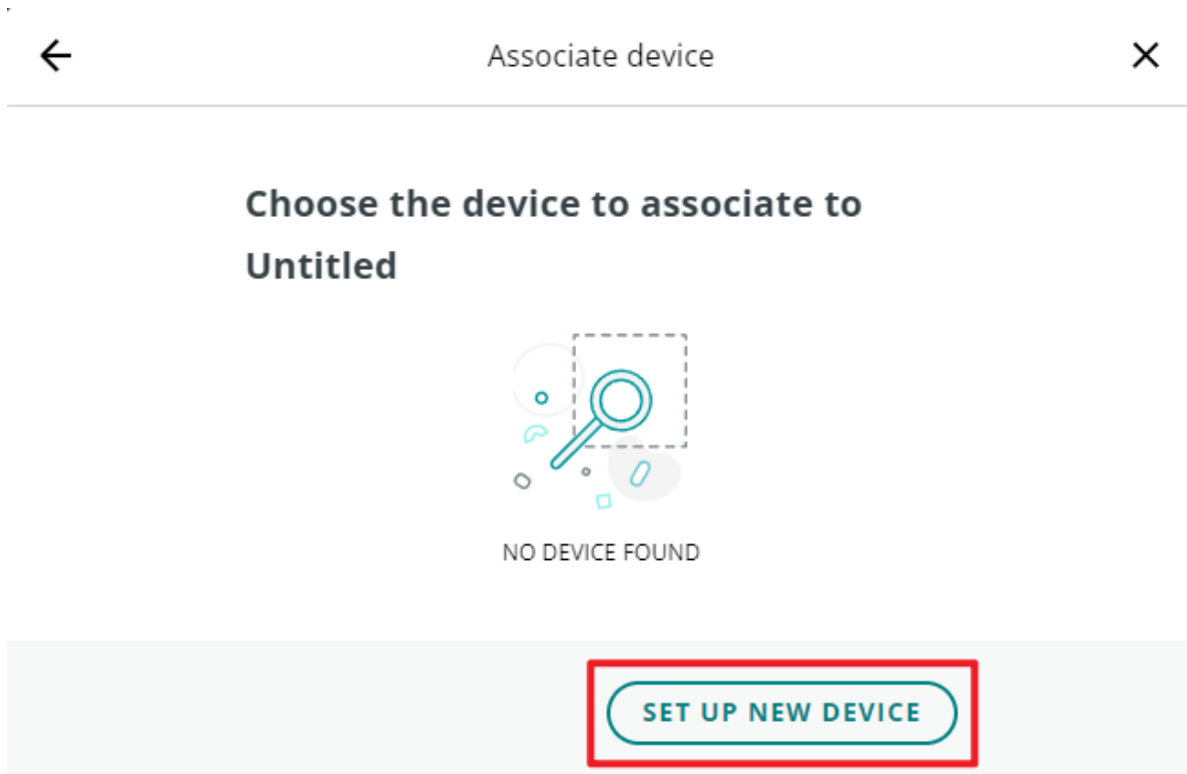
3. Erstellen Sie ein neues „Thing“ (Objekt).



4. Verknüpfen Sie Ihr Gerät.



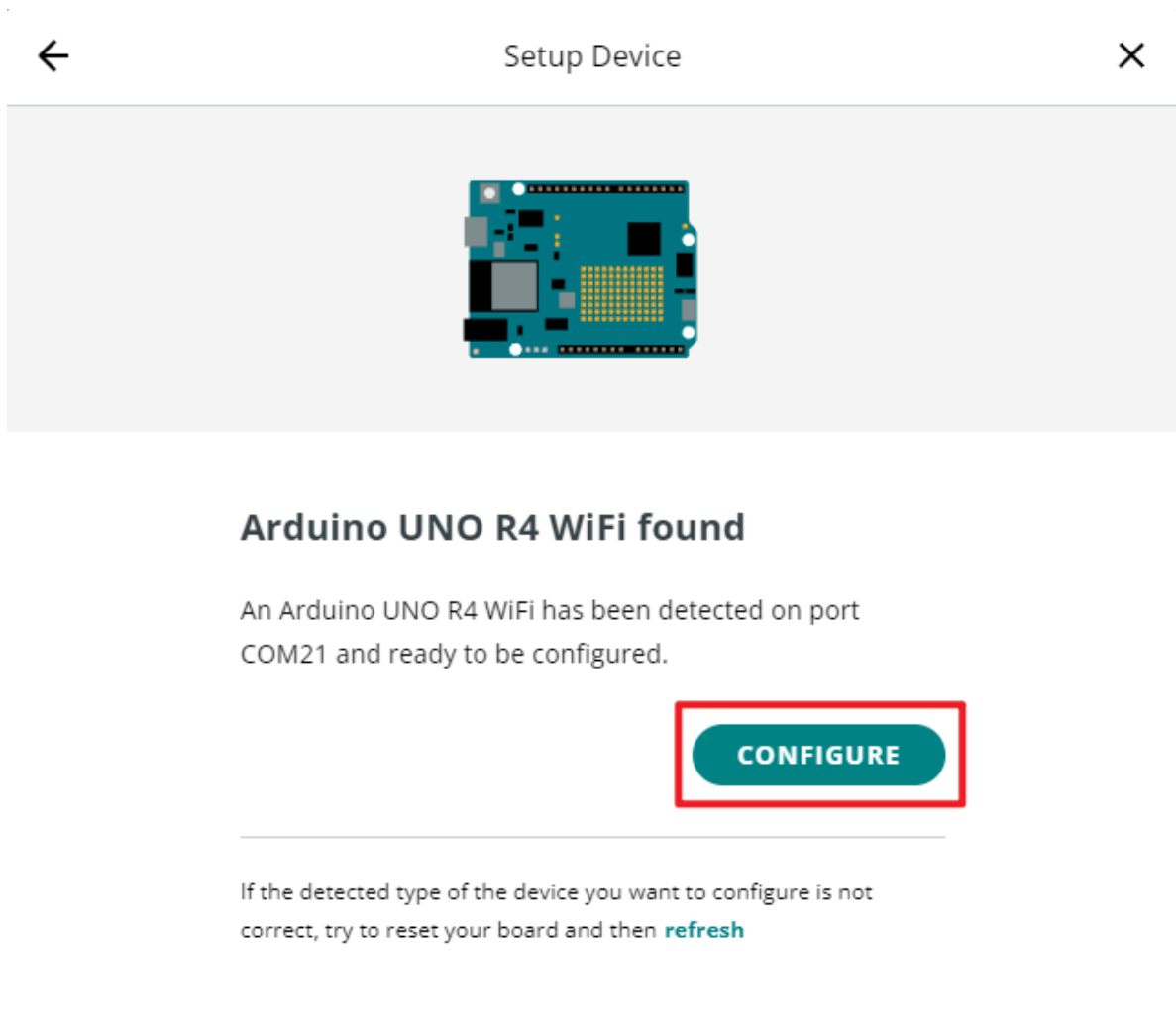
5. Richten Sie ein neues Gerät ein.



6. Wählen Sie Ihr Arduino-Board aus.

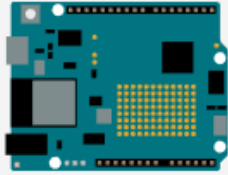


7. Warten Sie einen Moment, bis Ihr UNO R4 WiFi erkannt wird. Fahren Sie fort, indem Sie auf Konfigurieren klicken.



8. Geben Sie Ihrem Gerät einen Namen.

←Setup Device×



Give your device a name

Name your device so you will be able to recognize it.

Device Name


sunfounder

↻

NEXT

9. Machen Sie Ihr Gerät IoT-fähig und denken Sie daran, den geheimen Schlüssel zu speichern.

Setup Device ×




Make your device IoT-ready

To use this board you will need a Device ID and a Secret Key, please copy and save them or [download the PDF](#).

Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

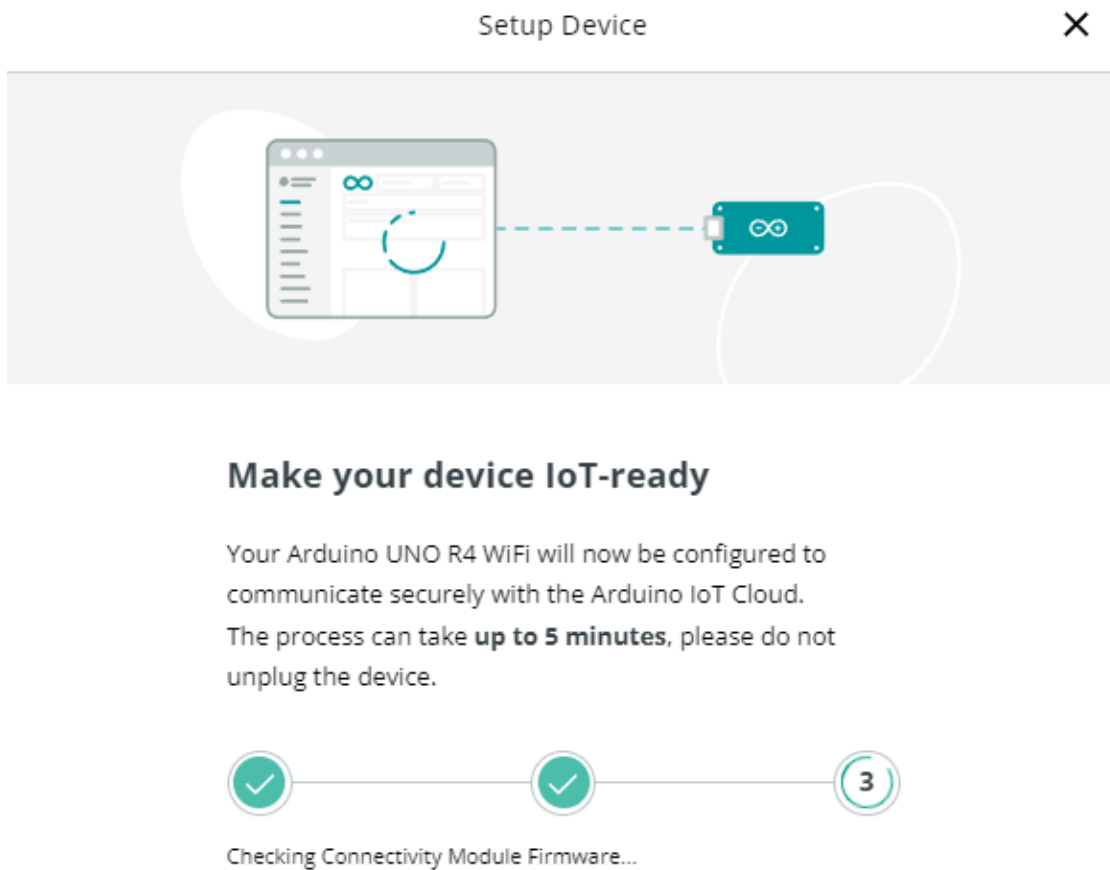
Device ID
fafc3977-b9a3-477b-ac53-7581ed104075

Secret Key
YGC9HYWNVXL1CGRUYB38

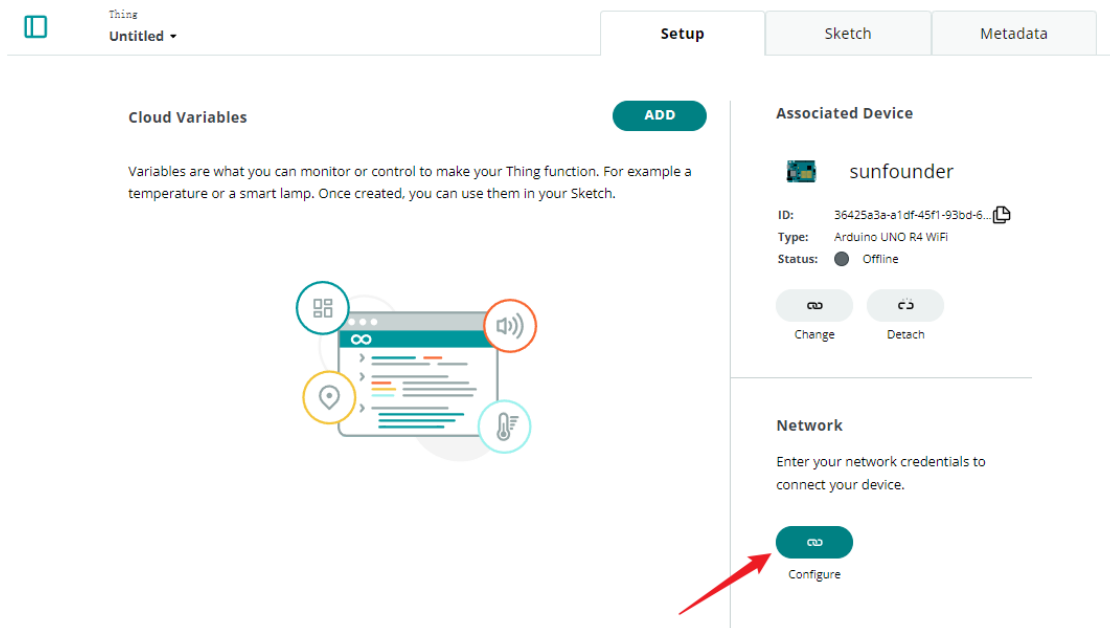
**Secret key cannot be recovered**
Please keep it safe, if you lose it you will have to delete and setup your device again.

☒ I saved my device ID and Secret Key CONTINUE

10. Warten Sie einige Minuten.



11. Konfigurieren Sie WiFi.



12. Hier müssen Sie Ihr WiFi-Passwort und den geheimen Schlüssel eingeben.

Configure network



You will find these network parameters in the secret tab in your sketch, and your device will be able to connect to the network once the sketch will be uploaded.

Wi-Fi Name *
MakerStarsHall

Password
.....

Secret Key *
.....

13. Fügen Sie eine Variable hinzu.

Thing
Untitled ▾

Setup
Sketch
Metadata

Cloud Variables

ADD

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

Associated Device

sunfounder

ID: 36425a3a-e1df-45f1-93bd-6...
Type: Arduino UNO R4 WiFi
Status: Offline

Change
Detach

Network

Wi-Fi Name: MakerSt...
Password:
Secret Key:

Change

14. Hier möchten wir die Temperatur in der IoT Cloud anzeigen, also konfigurieren wir eine schreibgeschützte Gleitkommavariablen.

Add variable

×

Name

Temperature

↻

Sync with other Things

i

Floating Point Number

eg. 1.55

▼

Declaration

`float temperature ;`

i

Variable Permission

i

☐ Read & Write

☒ Read Only

Variable Update Policy

i

☐ On change

☒ Periodically

Every

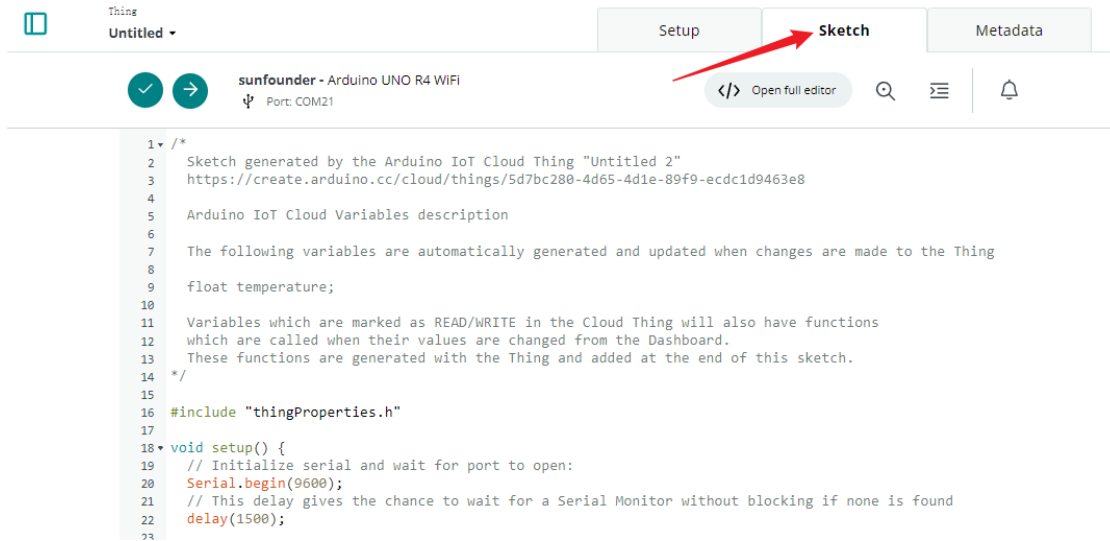
10

s

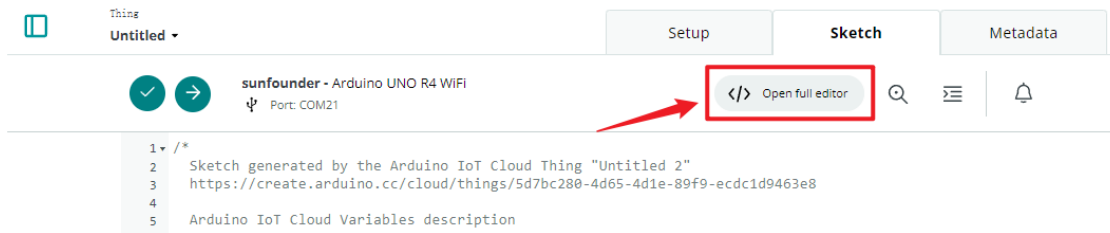
CANCEL

ADD VARIABLE

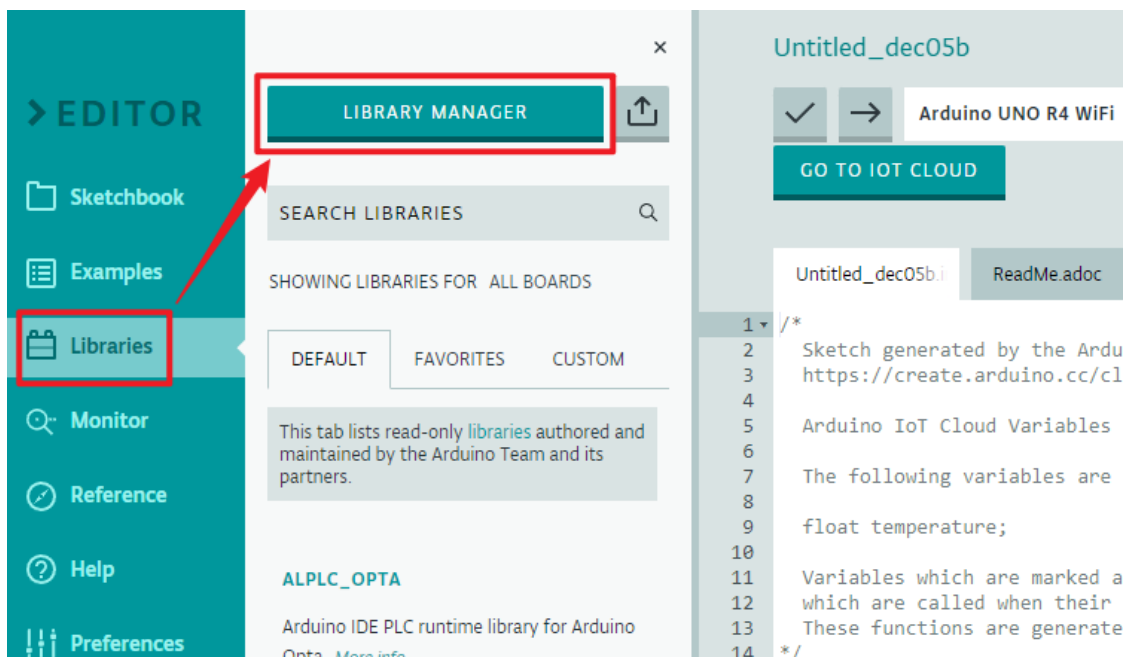
15. Nach Fertigstellung gehen Sie zum Sketch.



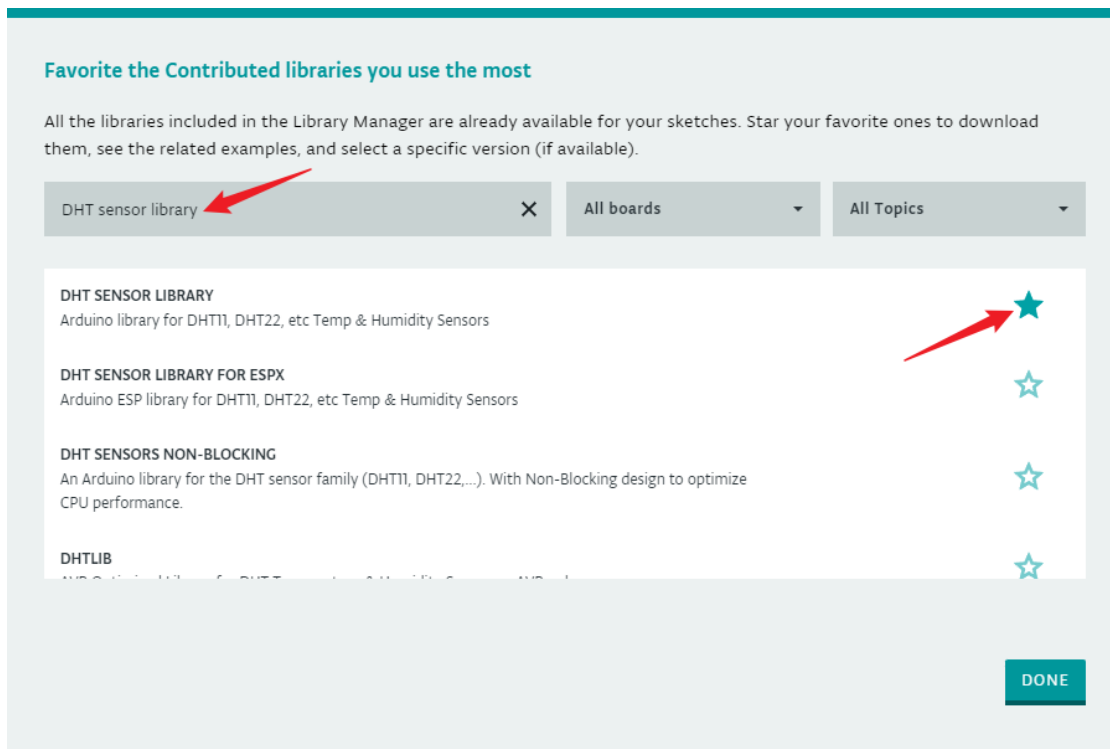
16. Öffnen Sie den Voll-Editor.



17. Klicken Sie auf der rechten Seite auf Bibliotheken, dann auf Bibliotheks-Manager.



18. Suchen Sie die DHT-Sensorbibliothek und markieren Sie sie.



19. Jetzt müssen wir den Code bearbeiten. Sie sehen, dass der Editor bereits den IoT Cloud-bezogenen Code für Sie vorbereitet hat. Sie müssen nur die spezifische Funktionalität hinzufügen, die Sie benötigen. In diesem Beispiel haben wir Code hinzugefügt, um die Temperatur mit dem DHT11-Sensor zu lesen.

```
// DHT sensor library - Version: Latest
#include <DHT.h>
#include <DHT_U.h>

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/260edac8-34f9-4e2e-9214-ba0c20994220

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes
are made to the Thing

float temperature;

Variables which are marked as READ/WRITE in the Cloud Thing will also have
functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this
sketch.
*/

#include "thingProperties.h"

#define DHTPIN 11
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without
  ↪ blocking if none is found
  delay(1500);

  dht.begin();

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

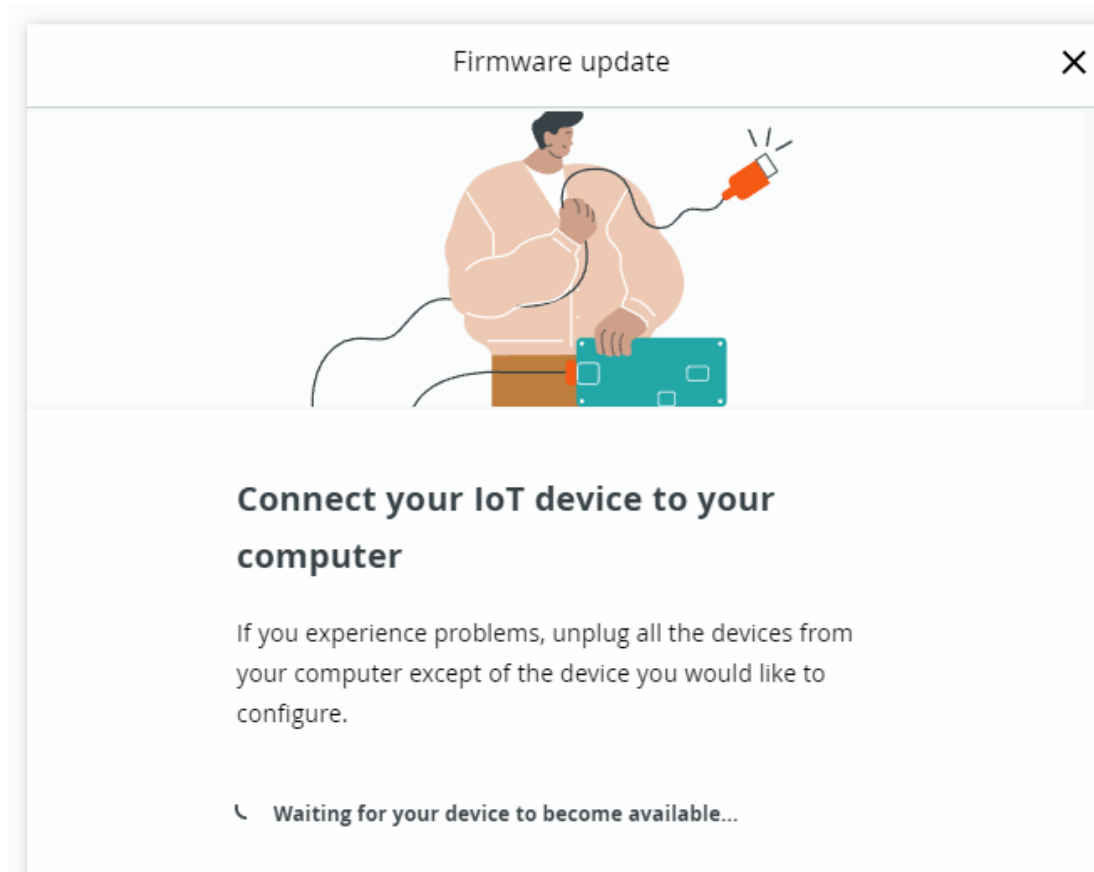
  /*
   * The following function allows you to obtain more information
   * related to the state of network and IoT Cloud connection and errors
   * the higher number the more granular information you'll get.
   * The default is 0 (only errors).
   * Maximum is 4
   */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here

  float temp = dht.readTemperature();
  temperature = temp;
}

```

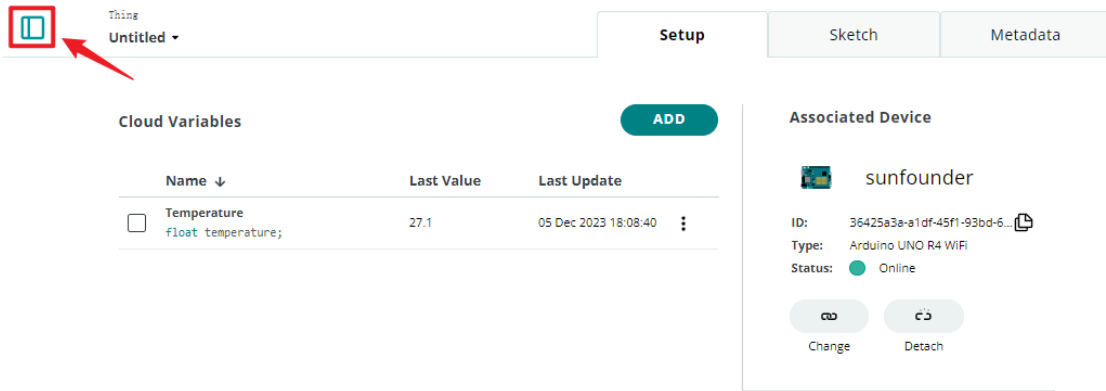
20. Laden Sie den Code hoch. Es könnte eine Aufforderung zur Aktualisierung erscheinen; folgen Sie den Anweisungen, um den Vorgang abzuschließen.



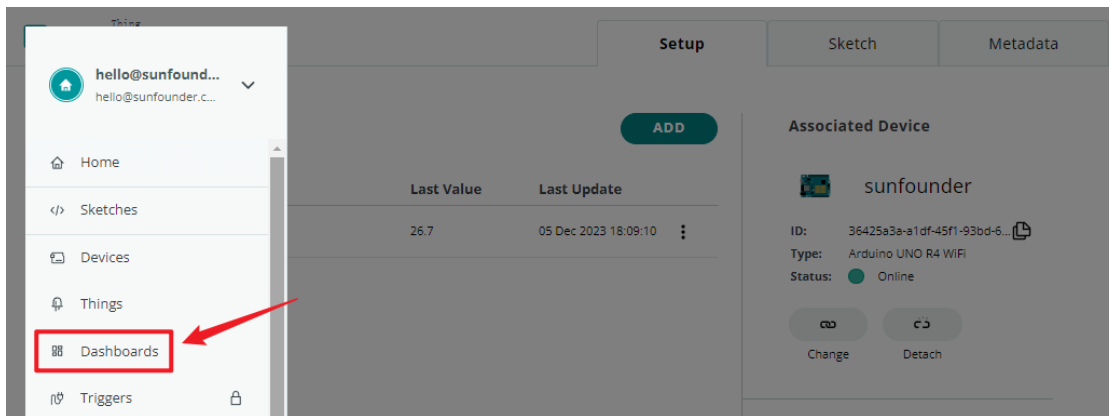
21. Kehren Sie zum IoT CLOUD zurück.



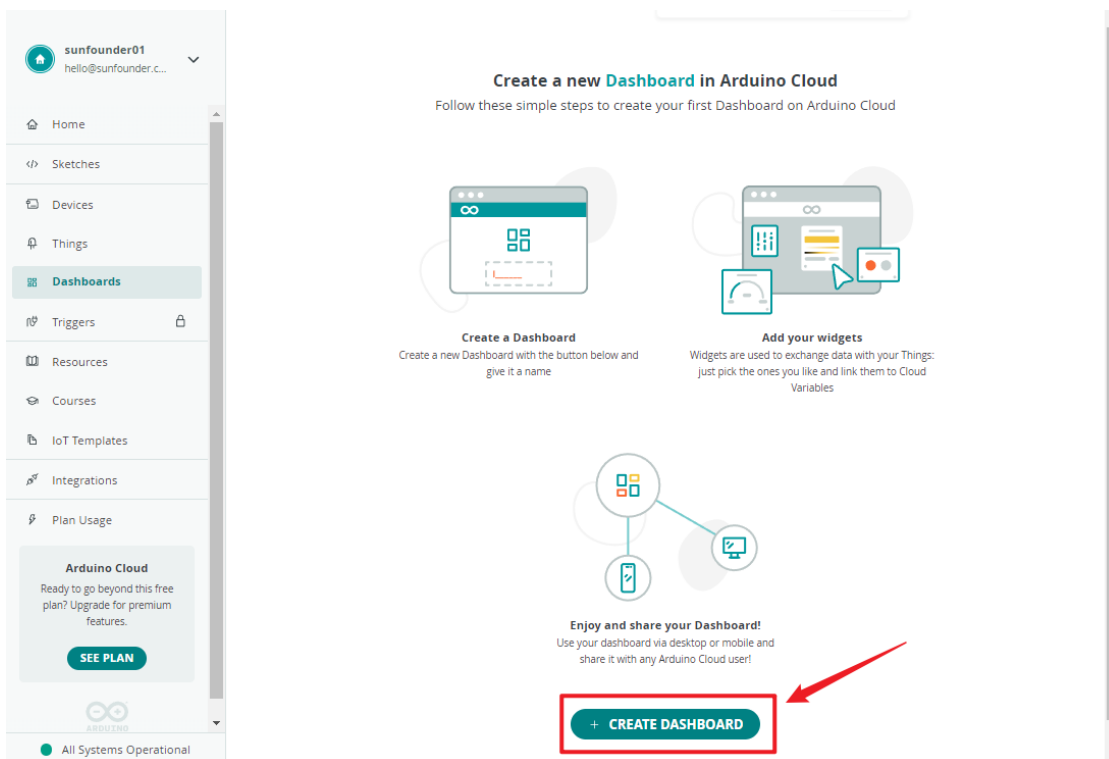
22. Klicken Sie auf das Menü in der oberen linken Ecke.



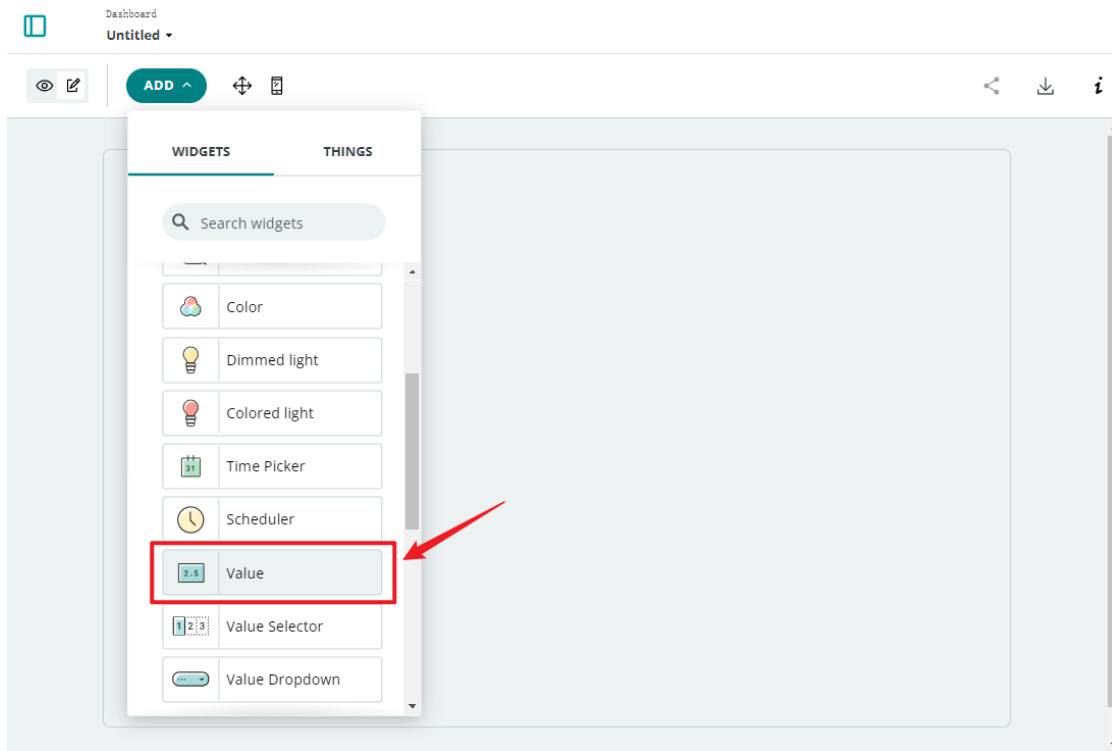
23. Klicken Sie auf das Dashboard.



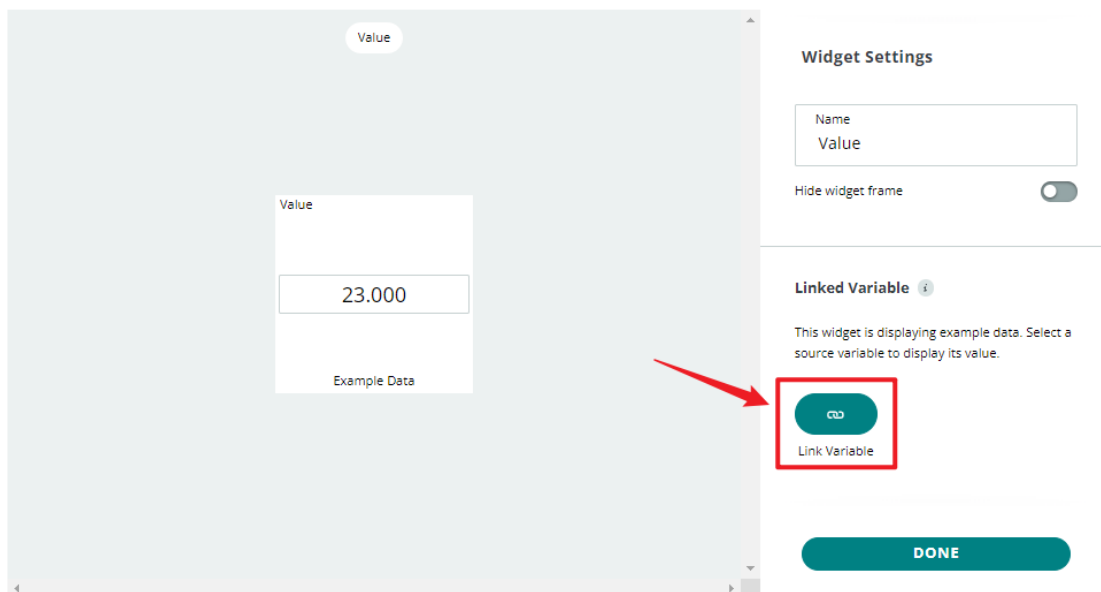
24. Erstellen Sie ein Dashboard.



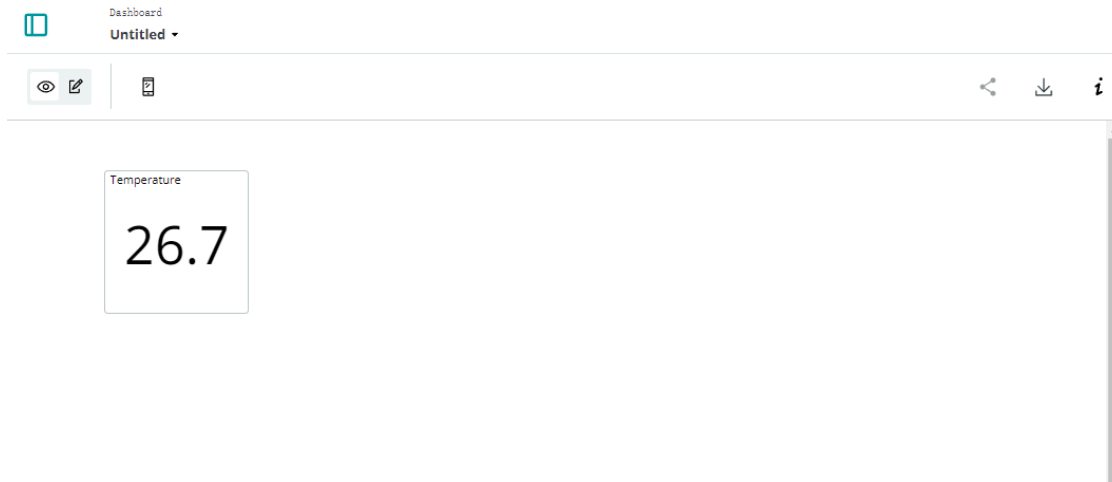
25. Es stehen viele Widgets zur Verfügung; hier wählen wir ein Wert-Widget zur Anzeige der Temperatur.



26. Nach dem Klicken erscheint eine Widget-Einstellungsfläche, auf der Sie das Widget mit der zuvor erstellten Cloud-Variablen verbinden können.



27. Nun können Sie die Sensordaten im Arduino IoT Cloud einsehen.



Wie funktioniert das?

Nachdem Sie die IoT Cloud konfiguriert haben (Geräteeinrichtung, Netzwerkeinrichtung, Erstellung von Cloud-Variablen), werden Sie feststellen, dass sich der Sketch in der Cloud automatisch aktualisiert. Daher ist der Großteil des Codes bereits für Sie geschrieben.

Öffnen Sie den Editor, und Sie werden sehen, dass dieser Sketch aus vier Dateien besteht:

main.ino: Wird verwendet, um das Arduino zu initialisieren und die Hauptaufgaben der Schleife auszuführen. Zusätzlich beinhaltet es die Logik für die Verbindung und Kommunikation mit der Arduino IoT Cloud.

thingProperties.h: Diese Datei wird verwendet, um Variablen und Funktionen in der Arduino IoT Cloud zu definieren. Sie enthält Deklarationen von Cloud-Variablen und deren zugehörigen Rückruffunktionen. Im bereitgestellten Code wird sie verwendet, um Cloud-Eigenschaften zu initialisieren (z. B. die Temperaturvariable) und eine Verbindung zur Arduino IoT Cloud herzustellen.

Secret: Wird verwendet, um sensible oder private Informationen wie WLAN-Passwörter oder API-Schlüssel zu speichern. Diese sensiblen Informationen werden typischerweise nicht direkt im Code offengelegt, sondern im Secret-File gespeichert, um die Sicherheit zu erhöhen.

ReadMe.adoc: Enthält Projektdokumentation oder andere relevante Informationen für ein einfacheres Verständnis und die Nutzung des Projekts. Diese Datei enthält normalerweise keinen ausführbaren Code, sondern Dokumente und beschreibende Informationen.

Wir müssen etwas Code für den DHT11-Sensor hinzufügen. Dieser Code ist identisch mit dem, den Sie in Ihrer lokalen IDE verwenden würden. Der einzige Unterschied besteht darin, dass Sie den vom DHT11 gelesenen Wert der Cloud-Variablen `temperature` zuweisen müssen.

(Hinweis: Sie sollten niemals `thingProperties.h` und `Secret` ändern. Sie werden geändert, wenn Sie Änderungen mit dem Thing-Editor vornehmen.)

7.3 Sicherheitssystem über IFTTT

Mit diesem Projekt erstellen wir ein Sicherheitsgerät, das einen PIR-Sensor verwendet, um Eindringlinge oder streunende Tiere, die Ihr Zuhause betreten, zu erkennen. Im Falle eines Einbruchs erhalten Sie eine E-Mail-Benachrichtigung.

Wir nutzen Webhooks als grundlegenden Dienst. Eine POST-Anfrage wird von UNO R4 an den IFTTT-Dienst gesendet.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

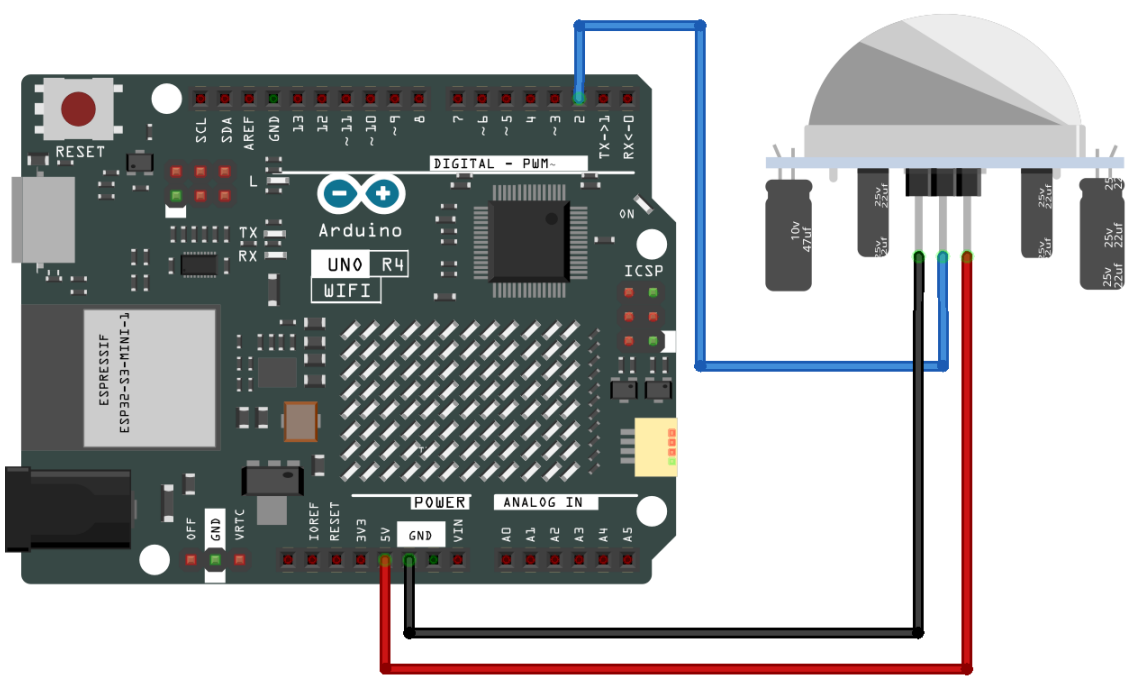
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

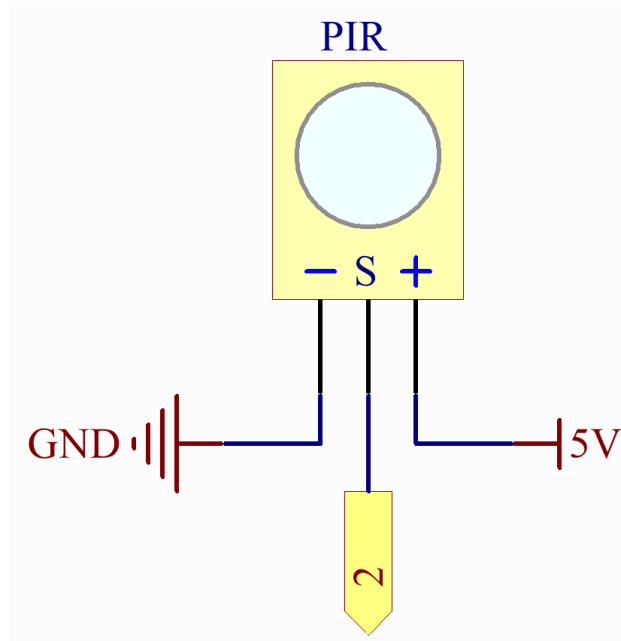
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Jumperkabel	
PIR-Bewegungssensormodul	

Verdrahtung



Schaltplan



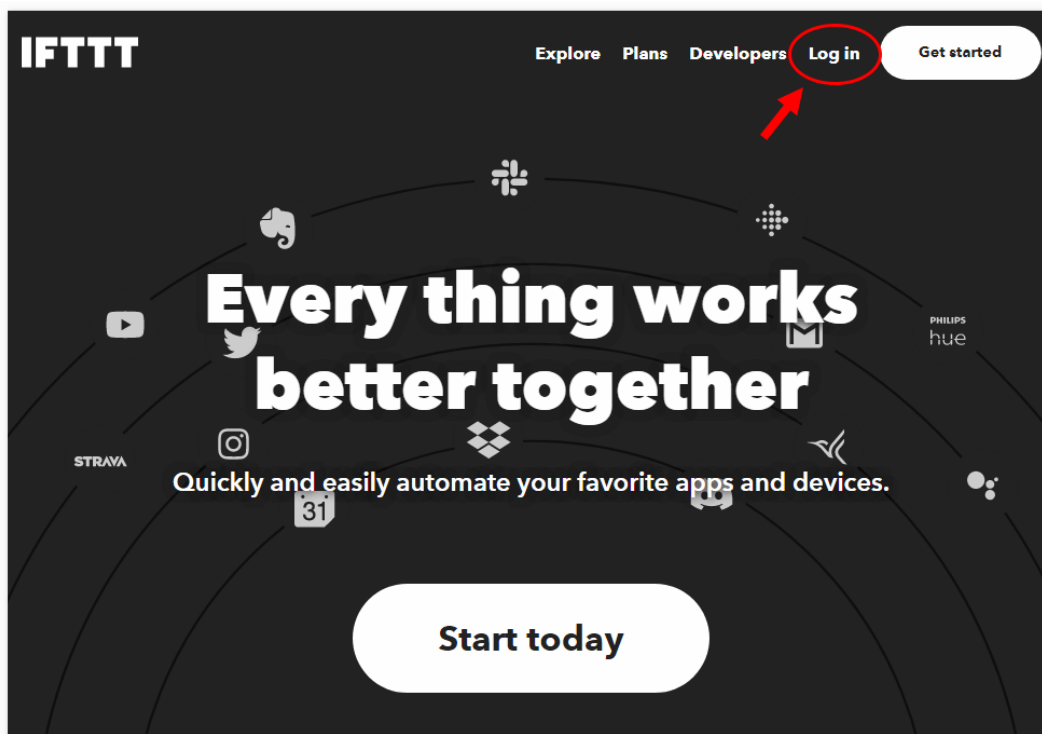
IFTTT einrichten

IFTTT ist ein kostenloser Dienst, der verschiedene Methoden bietet, um verschiedene Datendienste miteinander zu verknüpfen.

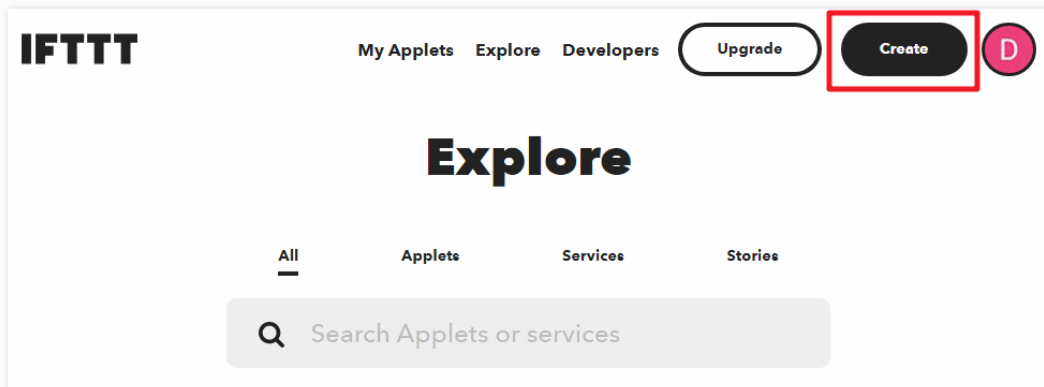
Erstellen wir ein Applet, das auf einen Webhook (benutzerdefinierte URL) reagiert, der Daten an IFTTT sendet, welcher Ihnen dann eine E-Mail sendet.

Bitte folgen Sie den untenstehenden Schritten auf IFTTT.

1. Besuchen Sie <https://ifttt.com>, um sich anzumelden oder ein Konto zu erstellen.



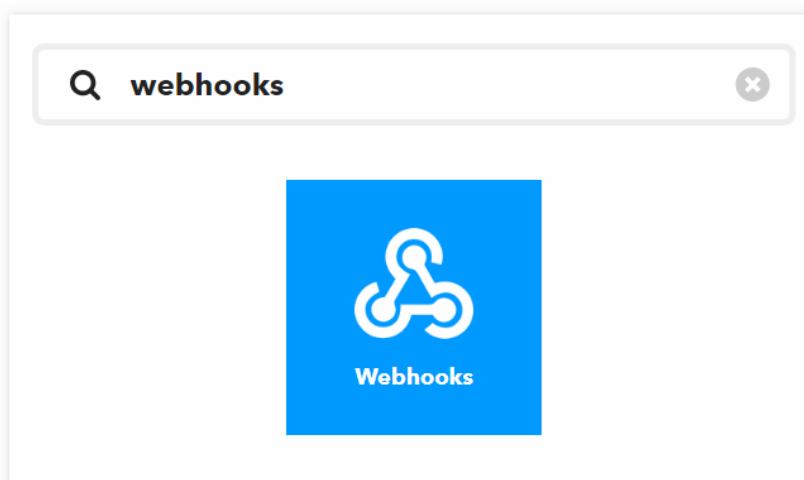
2. Klicken Sie auf **Create**.



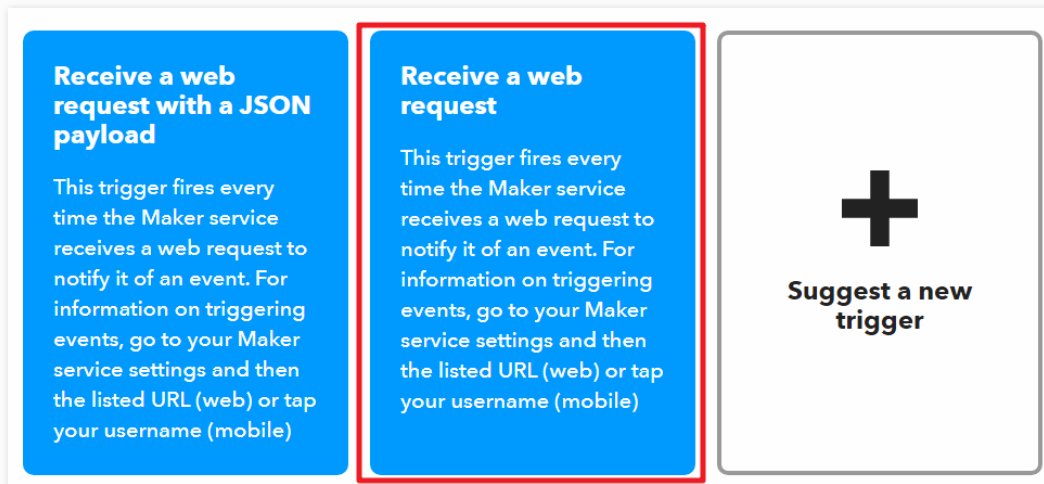
3. Fügen Sie ein **If This** Ereignis hinzu.



4. Suchen Sie nach **Webhooks**.



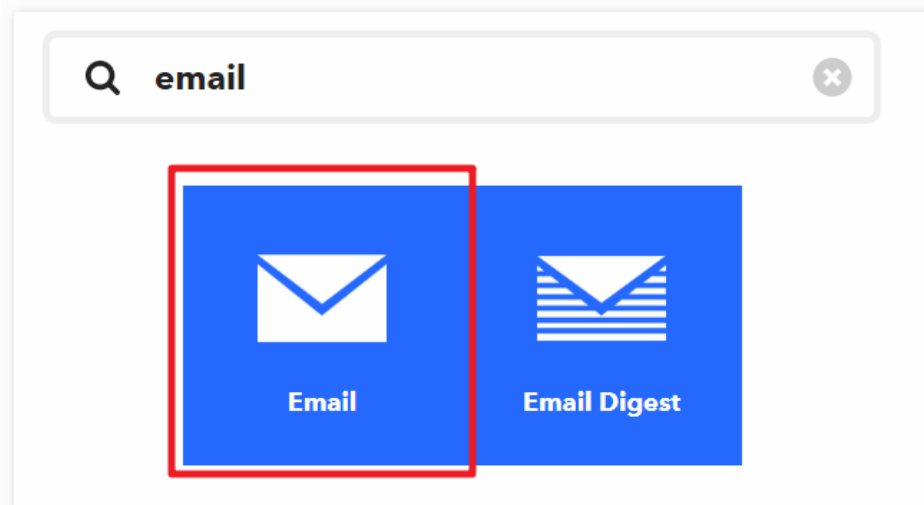
5. Wählen Sie **Receive a web request**.



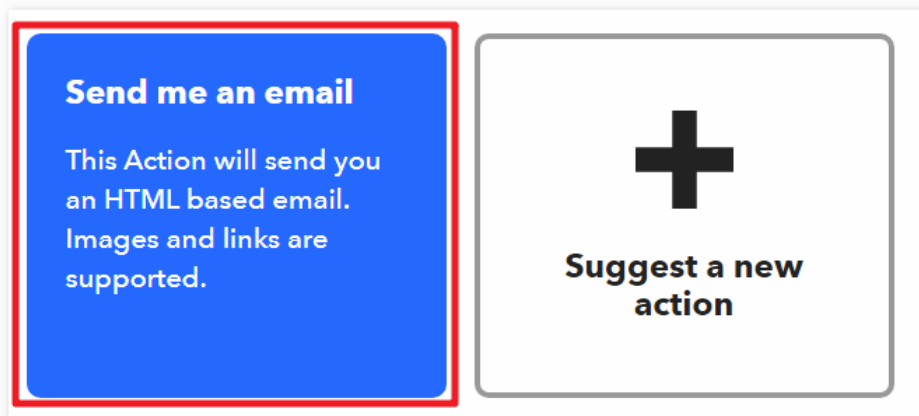
6. Geben Sie den Ereignisnamen ein (z. B. SecurityWarning) und klicken Sie auf **Create trigger**.

7. Fügen Sie ein **Then That** Ereignis hinzu.


8. Suchen Sie nach E-Mail.



9. Wählen Sie **Send me an email**.



10. Geben Sie den **Subject** und den **Body** ein, dann klicken Sie auf **Create action**.



Send me an email

This Action will send you an HTML based email. Images and links are supported.

Subject

SecurityWarning

Add ingredient

Body

What: EventName

When: OccurredAt

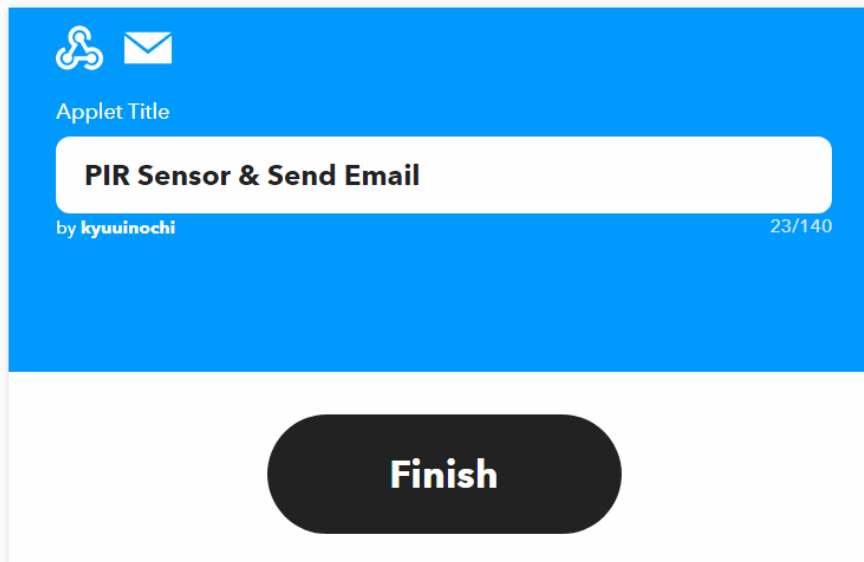
Add ingredient

Create action

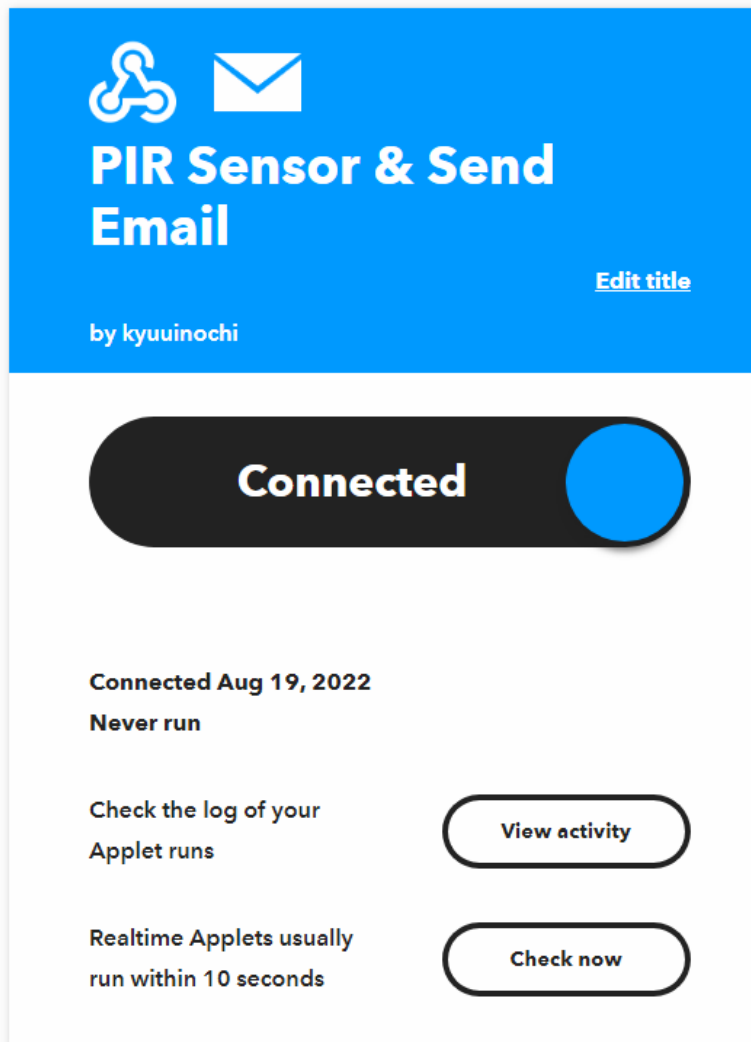
11. Klicken Sie auf **Continue** um die Einrichtung abzuschließen.



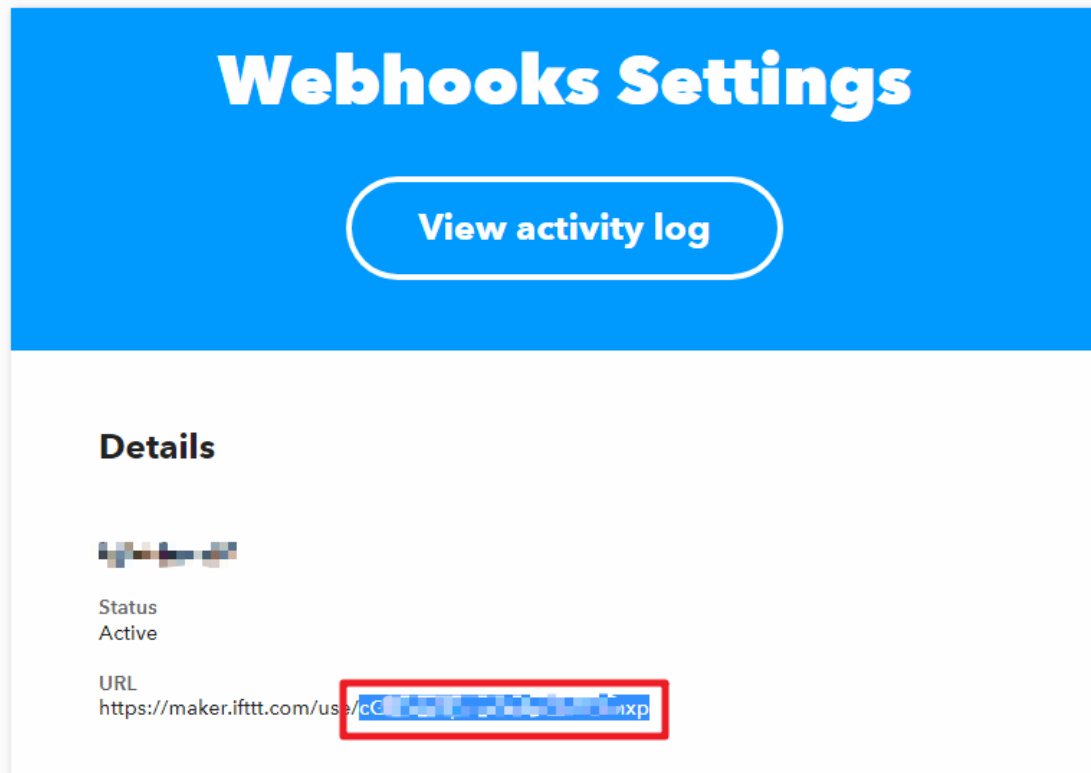
12. Passen Sie nach Bedarf den Titelnamen an.



13. Sie werden automatisch zur Seite mit den Applet-Details weitergeleitet, wo Sie sehen können, dass das Applet derzeit verbunden ist und Sie den Schalter zum Aktivieren/Deaktivieren umlegen können.



14. Nachdem wir das IFTTT-Applet erstellt haben, benötigen wir auch den Webhooks-Schlüssel, den Sie von erhalten können, damit Ihr Gerät auf IFTTT zugreifen kann.



15. Kopieren Sie den Webhooks-Schlüssel in „arduino_secrets.h“ und tragen Sie Ihr SSID und Passwort ein.

```
#define SECRET_SSID "your_ssid"           // your network SSID (name)
#define SECRET_PASS "your_password"       // your network password (used for WPA,
// or as a key for WEP)
#define WEBHOOKS_KEY "your_key"
```

Den Code ausführen

Bemerkung:

- Sie können die Datei 03_ifttt_pir.ino direkt im Pfad elite-explorer-kit-main\iot_project\03_ifttt_pir öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Im Code werden SSID und Passwort in `arduino_secrets.h` gespeichert. Bevor Sie dieses Beispiel hochladen, müssen Sie sie mit Ihren eigenen WLAN-Anmeldeinformationen ändern. Außerdem sollten Sie aus Sicherheitsgründen diese Informationen vertraulich halten, wenn Sie den Code teilen oder speichern.

Warnung: Um zu verhindern, dass Ihr Postfach überflutet wird, debuggen Sie bitte den *PIR-Bewegungssensormodul* vorher, bevor Sie den Code für dieses Projekt ausführen.

Wie funktioniert des?

1. Einbinden der erforderlichen Bibliotheken und Header-Dateien:

- "WiFiS3.h": Wird verwendet, um Wi-Fi-Verbindungen zu verwalten.
- "arduino_secrets.h": Enthält den Namen und das Passwort des Wi-Fi-Netzwerks, um sensible Informationen zu schützen.

2. Definition einiger globaler Variablen und Konstanten:

- `ssid`: Name des Wi-Fi-Netzwerks.
- `pass`: Wi-Fi-Netzwerkpasswort.
- `status`: Status der Wi-Fi-Verbindung.
- `client`: Client zur Kommunikation mit dem Wi-Fi-Server.
- `server`: Adresse des IFTTT Webhook-Servers.
- `event`: Name des IFTTT Webhook-Ereignisses.
- `webRequestURL`: Konstruierte URL für das Senden von HTTP-Anfragen, einschließlich des Webhook-Ereignisnamens und -schlüssels.
- `pirPin`: Digitaler Pin, an den der PIR-Sensor angeschlossen ist.
- `motionDetected`: Flag-Variable zur Verfolgung der Bewegungserkennung.

3. `setup()` Funktion:

- Initialisiert die serielle Kommunikation.
- Überprüft das Vorhandensein des Wi-Fi-Moduls und gibt dessen Firmware-Version aus.
- Versucht, eine Verbindung zum Wi-Fi-Netzwerk herzustellen, mit Wiederholungen bei Misserfolg.
- Setzt den an den PIR-Sensor angeschlossenen Pin auf Eingabemodus.

4. `readResponse()` Funktion:

- Liest HTTP-Antwortdaten vom IFTTT-Server und gibt sie auf der seriellen Konsole aus.

5. `loop()` Funktion:

- Ruft die Funktion `readResponse()` auf, um HTTP-Antwortdaten zu lesen.
- **Überprüft die Bewegung mithilfe des PIR-Sensors. Wenn eine Bewegung erkannt wird und zuvor nicht erkannt wurde:**
 - Gibt „Motion detected!“ auf der Konsole aus.
 - Ruft die Funktion `triggerIFTTTEvent()` auf, um eine HTTP-Anfrage an den IFTTT-Server zu senden und das Webhook-Ereignis auszulösen.
 - Setzt das Flag `motionDetected` auf `true`, um anzuzeigen, dass eine Bewegung erkannt wurde.
- Wenn keine Bewegung erkannt wird, setzt das Flag `motionDetected` auf `false`.

6. `triggerIFTTTEvent()` Funktion:

- Stellt eine Verbindung mit dem IFTTT-Server her.
- Sendet eine HTTP-GET-Anfrage, einschließlich der URL des Webhook-Ereignisses und anderer HTTP-Header.

7. `printWifiStatus()` Funktion:

- Gibt Informationen über das verbundene Wi-Fi-Netzwerk aus, einschließlich SSID, IP-Adresse und Signalstärke (RSSI) auf der seriellen Konsole.

7.4 Cloud-Anrufsystem mit MQTT

Message Queuing Telemetry Transport (MQTT) ist ein einfaches Messaging-Protokoll und das am weitesten verbreitete in der Welt des Internet of Things (IoT).

MQTT-Protokolle definieren, wie IoT-Geräte Daten austauschen. Sie funktionieren ereignisgesteuert und sind über das Publish/Subscribe-Modell miteinander verbunden. Der Sender (Publisher) und der Empfänger (Subscriber) kommunizieren über Themen. Ein Gerät veröffentlicht eine Nachricht zu einem bestimmten Thema, und alle Geräte, die dieses Thema abonniert haben, erhalten die Nachricht.

In diesem Abschnitt erstellen wir ein Serviceklingelsystem mit UNO R4, HiveMQ (einem kostenlosen öffentlichen MQTT-Broker-Dienst) und vier Tasten. Jede der vier Tasten entspricht einem Restauranttisch, und wenn ein Kunde eine Taste drückt, können Sie auf HiveMQ sehen, welcher Tisch bedient werden muss.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

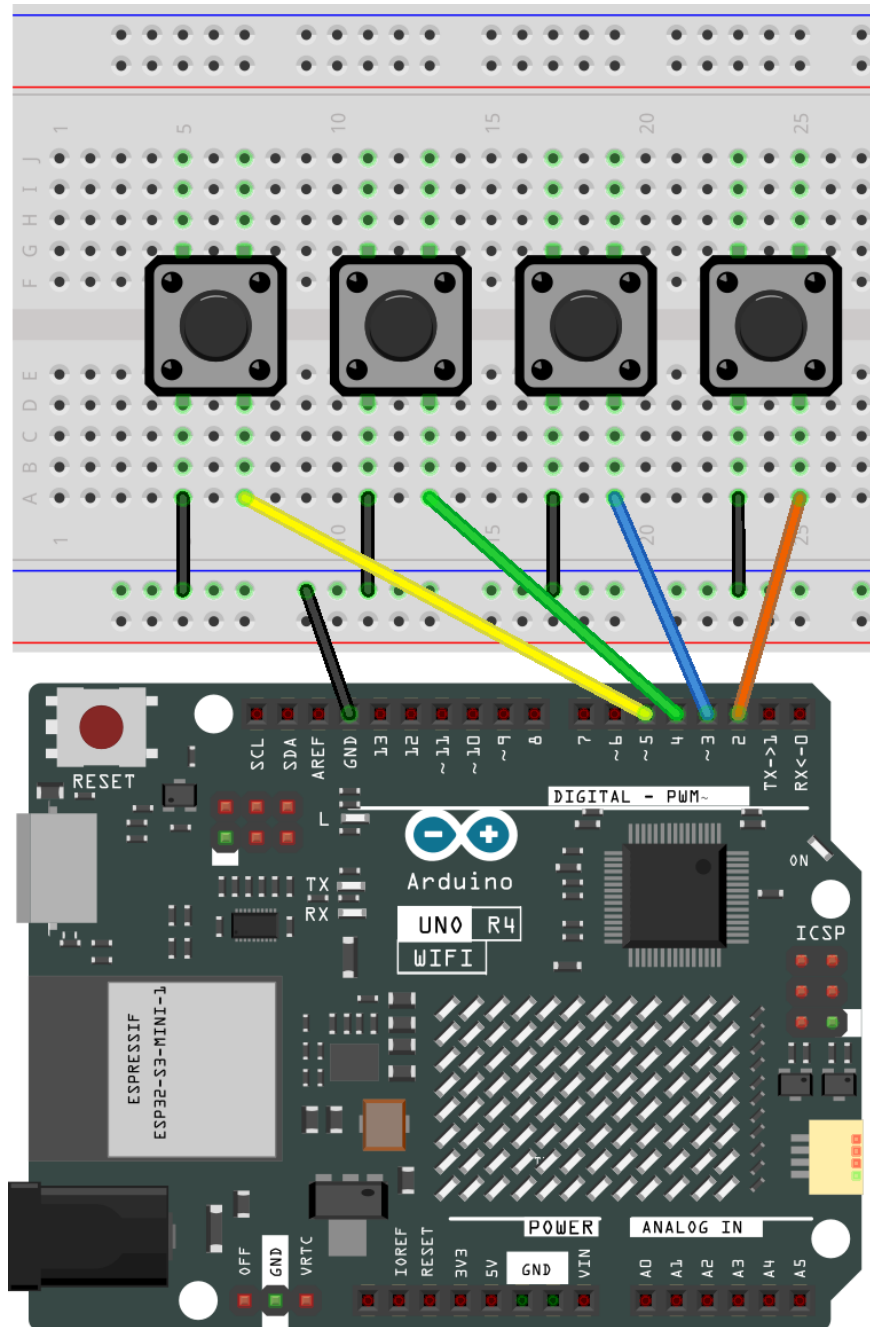
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

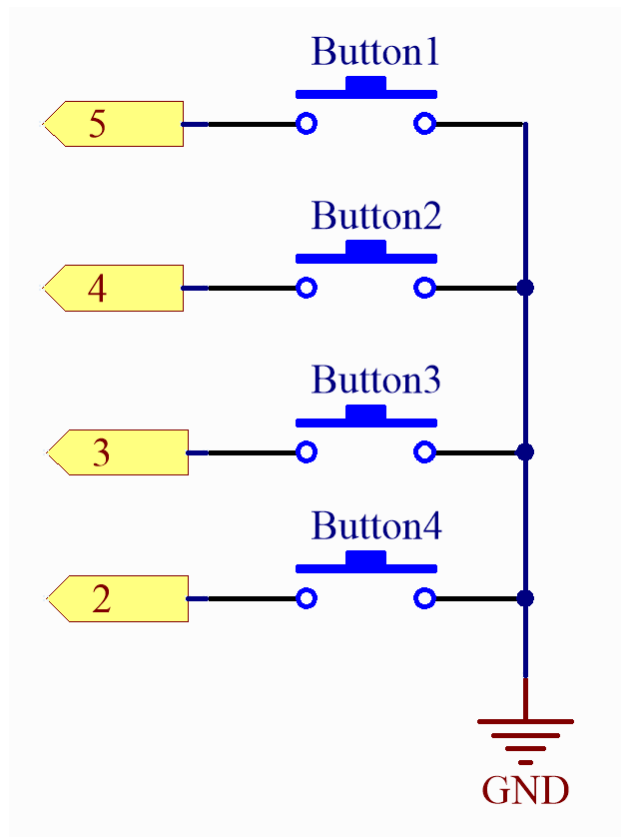
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Steckbrett</i>	
<i>Jumperkabel</i>	
<i>Knopf</i>	

Verdrahtung



Schaltplan



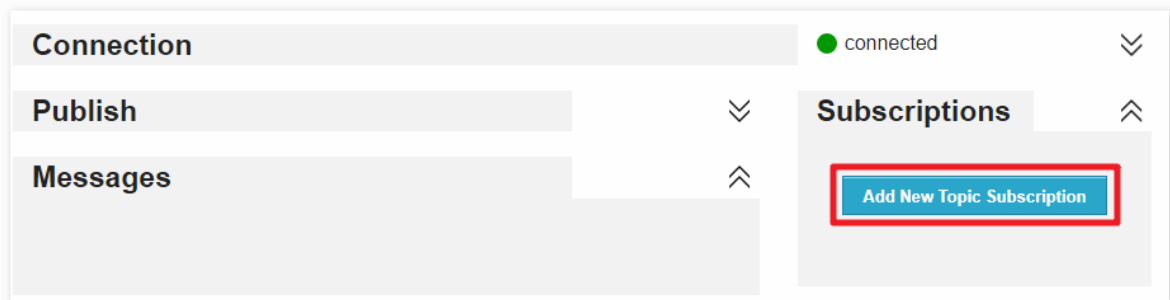
Wie funktioniert des?

HiveMQ ist ein MQTT-Broker und eine auf Client-Nachrichten basierende Plattform, die schnelle, effiziente und zuverlässige Datenübertragungen zu IoT-Geräten ermöglicht.

1. Öffnen Sie in Ihrem Webbrowser.
2. Verbinden Sie den Client mit dem standardmäßigen öffentlichen Proxy.

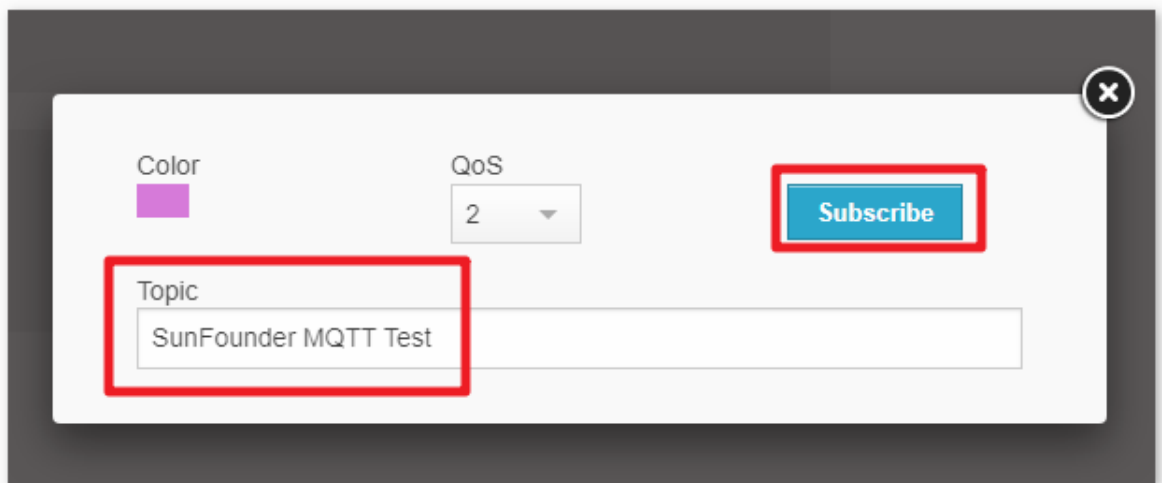
The screenshot shows the 'Connection' dialog box in the MQTT client interface. The status at the top right is 'disconnected'. The 'Host' field contains 'broker.mqttdashboard.com', the 'Port' is '8000', and the 'ClientID' is 'clientId-XINXbl3X1Q'. The 'Connect' button is highlighted with a red box. Other fields include 'Username', 'Password', 'Keep Alive' (60), 'SSL' (unchecked), 'Clean Session' (checked), 'Last-Will Topic', 'Last-Will QoS' (0), and 'Last-Will Retain' (unchecked).

3. Klicken Sie auf **Add New Topic Subscription**.



4. Geben Sie die Themen ein, denen Sie folgen möchten, und klicken Sie auf **Subscribe**. Stellen Sie sicher, dass die von Ihnen festgelegten Themen einzigartig sind, um Nachrichten von anderen Benutzern zu vermeiden, und achten Sie auf Groß- und Kleinschreibung.

In diesem Beispielcode haben wir das Thema als `SunFounder MQTT Test` festgelegt. Wenn Sie Änderungen vorgenommen haben, stellen Sie sicher, dass das Thema im Code mit dem auf der Webseite abonnierten Thema übereinstimmt.



Bibliothek installieren

Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „ArduinoMqttClient“, um sie zu installieren.

`ArduinoMqttClient.h`: Wird für die MQTT-Kommunikation verwendet.

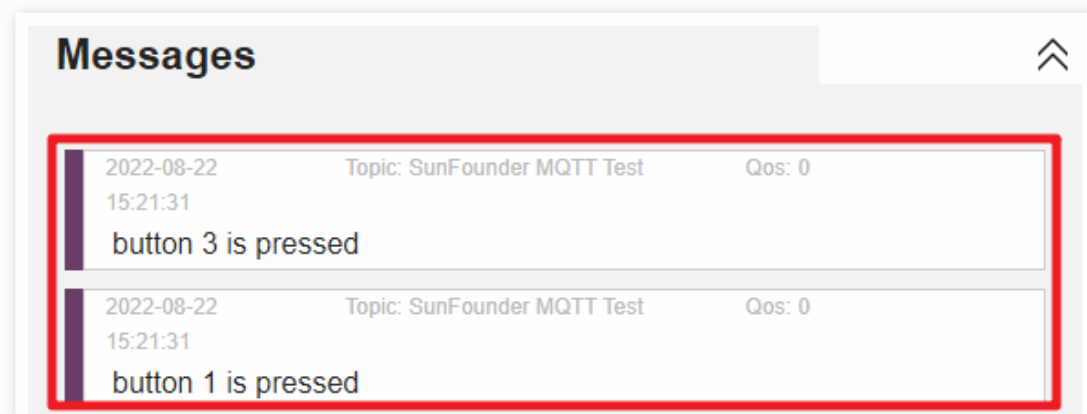
Code ausführen

Bemerkung:

- Sie können die Datei `04_mqtt_button.ino` direkt unter dem Pfad `elite-explorer-kit-main\iot_project\04_mqtt_button` öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Im Code werden SSID und Passwort in `arduino_secrets.h` gespeichert. Bevor Sie dieses Beispiel hochladen, müssen Sie sie mit Ihren eigenen WLAN-Anmeldeinformationen ändern. Außerdem sollten Sie aus Sicherheitsgründen diese Informationen vertraulich halten, wenn Sie den Code teilen oder speichern.

Nachdem Sie den Code ausgeführt haben, gehen Sie zurück zu , und wenn Sie einen der Tasten auf dem Breadboard drücken, sehen Sie die Nachrichtenaufforderung auf HiveMQ.



Wie funktioniert des?

Dieser Code ist für ein Arduino-basiertes Projekt, das sich mit Wi-Fi verbindet und über das MQTT-Protokoll mit einem MQTT-Broker kommuniziert. Zusätzlich kann es erkennen, ob vier Tasten gedrückt werden, und sendet die entsprechenden Nachrichten an den MQTT-Broker.

Hier ist eine detaillierte Erklärung des Codes:

1. Einschließen relevanter Bibliotheken:

```
#include <WiFiS3.h>
#include <ArduinoMqttClient.h>
```

2. Sensible Informationen einbinden:

- Die Datei `arduino_secrets.h` enthält den SSID und das Passwort für das Wi-Fi-Netzwerk.

```
#include "arduino_secrets.h"
char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
```

3. Variablen initialisieren:

- Variablen für die Verwaltung von Wi-Fi- und MQTT-Verbindungen.
- Initialisieren der Tastenpins und Tastenzustände.

4. setup():

- Initialisieren der seriellen Kommunikation.
- Überprüfen des Vorhandenseins des Wi-Fi-Moduls und Versuch, eine Verbindung zu Wi-Fi herzustellen.
- Netzwerkdaten ausgeben.
- Versuch, eine Verbindung zum MQTT-Broker herzustellen.
- Abonnieren von MQTT-Themen.
- Tasten auf Eingabemodus setzen.

5. loop():

- Die MQTT-Verbindung aktiv halten.

- Überprüfen, ob jede Taste gedrückt wird, und gegebenenfalls MQTT-Nachrichten senden.

6. Andere Hilfsfunktionen:

- `printWifiData()`: Gibt Informationen über das aktuell verbundene Wi-Fi-Netzwerk aus.
- `printCurrentNet()`: Gibt relevante Daten über das aktuelle Netzwerk aus.
- `printMacAddress(byte mac[])`: Gibt die MAC-Adresse aus.
- `onMqttMessage(int messageSize)`: Rückruffunktion, die ausgelöst wird, wenn eine Nachricht vom MQTT-Broker empfangen wird. Sie gibt das empfangene Nachrichtenthema und den Inhalt aus.
- `sendButtonMessage(int buttonNumber)`: Verwenden Sie diese Funktion, um MQTT-Nachrichten zu senden, wenn eine Taste gedrückt wird.

7.5 CherryLight

CheerLights ist ein globales Netzwerk synchronisierter Lichter, das von jedem gesteuert werden kann. Treten Sie der LED-Farbwechsel-Community bei, die es ermöglicht, dass LEDs weltweit gleichzeitig ihre Farben ändern. Platzieren Sie Ihre LEDs in einer Ecke Ihres Büros, um sich daran zu erinnern, dass Sie nicht alleine sind.

In diesem Fall nutzen wir ebenfalls MQTT, aber anstatt unsere eigenen Nachrichten zu veröffentlichen, abonnieren wir das Thema „cheerlights“. Dadurch können wir Nachrichten, die von anderen an das Thema „cheerlights“ gesendet werden, empfangen und diese Informationen verwenden, um die Farbe unseres LED-Streifens entsprechend zu ändern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

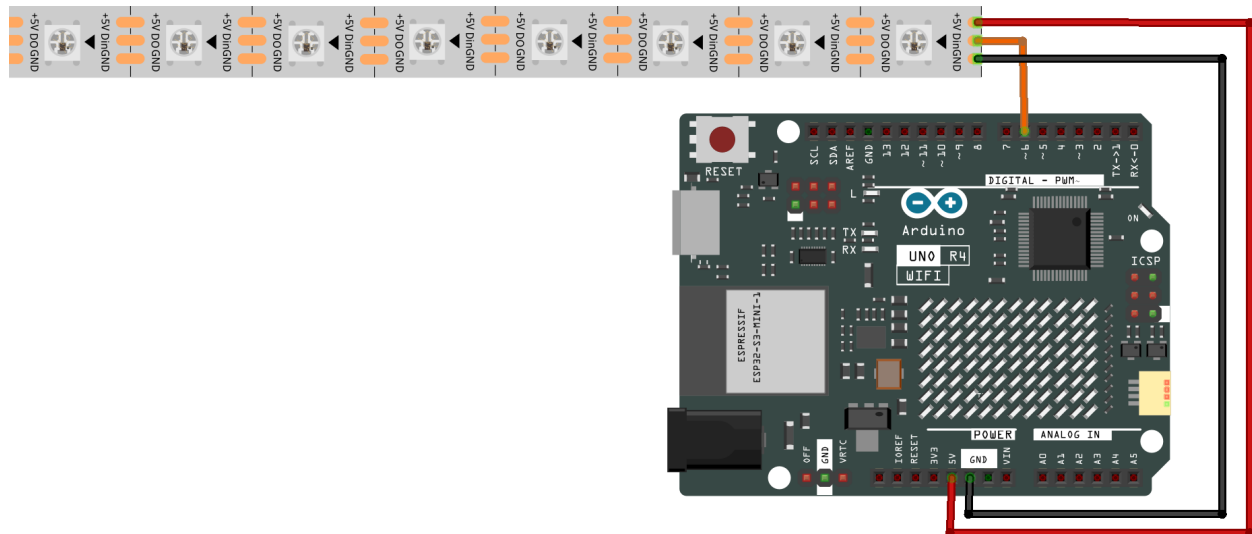
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

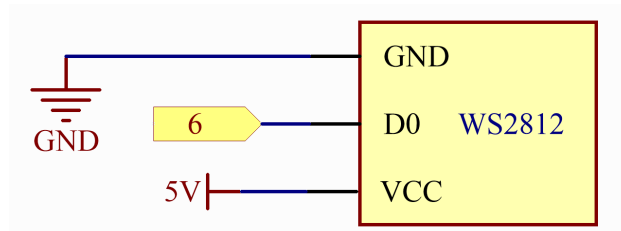
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 WiFi</i>	-
<i>Jumperkabel</i>	
<i>WS2812 RGB 8 LEDs Streifen</i>	

Verdrahtung



Schaltplan



Bibliothek installieren

Um die Bibliothek zu installieren, verwenden Sie den Arduino Library Manager und suchen Sie nach „ArduinoMqttClient“ und „FastLED“, um sie zu installieren.

ArduinoMqttClient.h: Wird für die MQTT-Kommunikation verwendet.

FastLED.h: Wird verwendet, um den RGB-LED-Streifen zu steuern.

Code ausführen

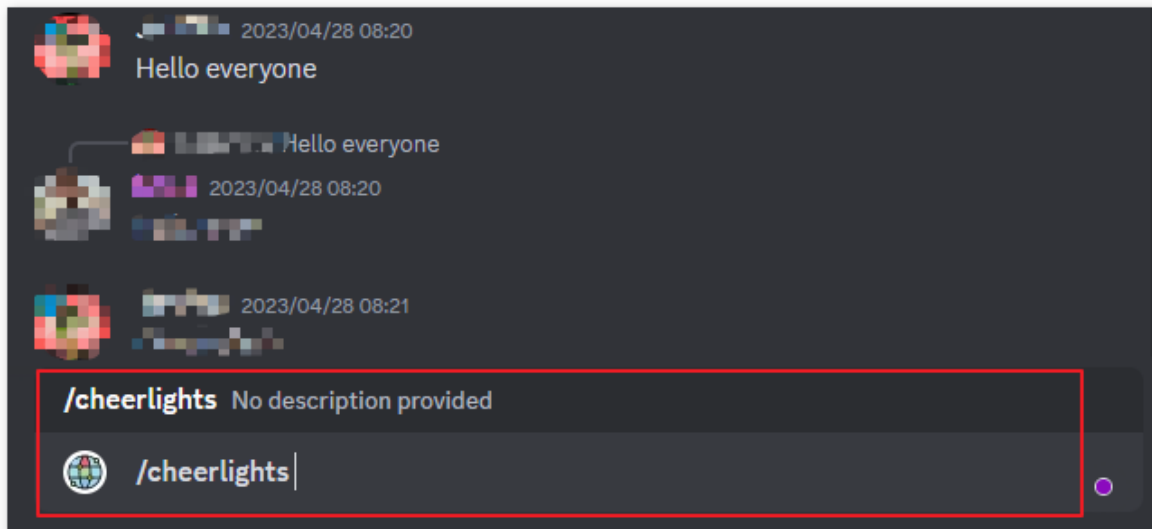
Bemerkung:

- Sie können die Datei 05_cheerlight.ino direkt unter dem Pfad elite-explorer-kit-main\iot_project\05_cheerlight öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

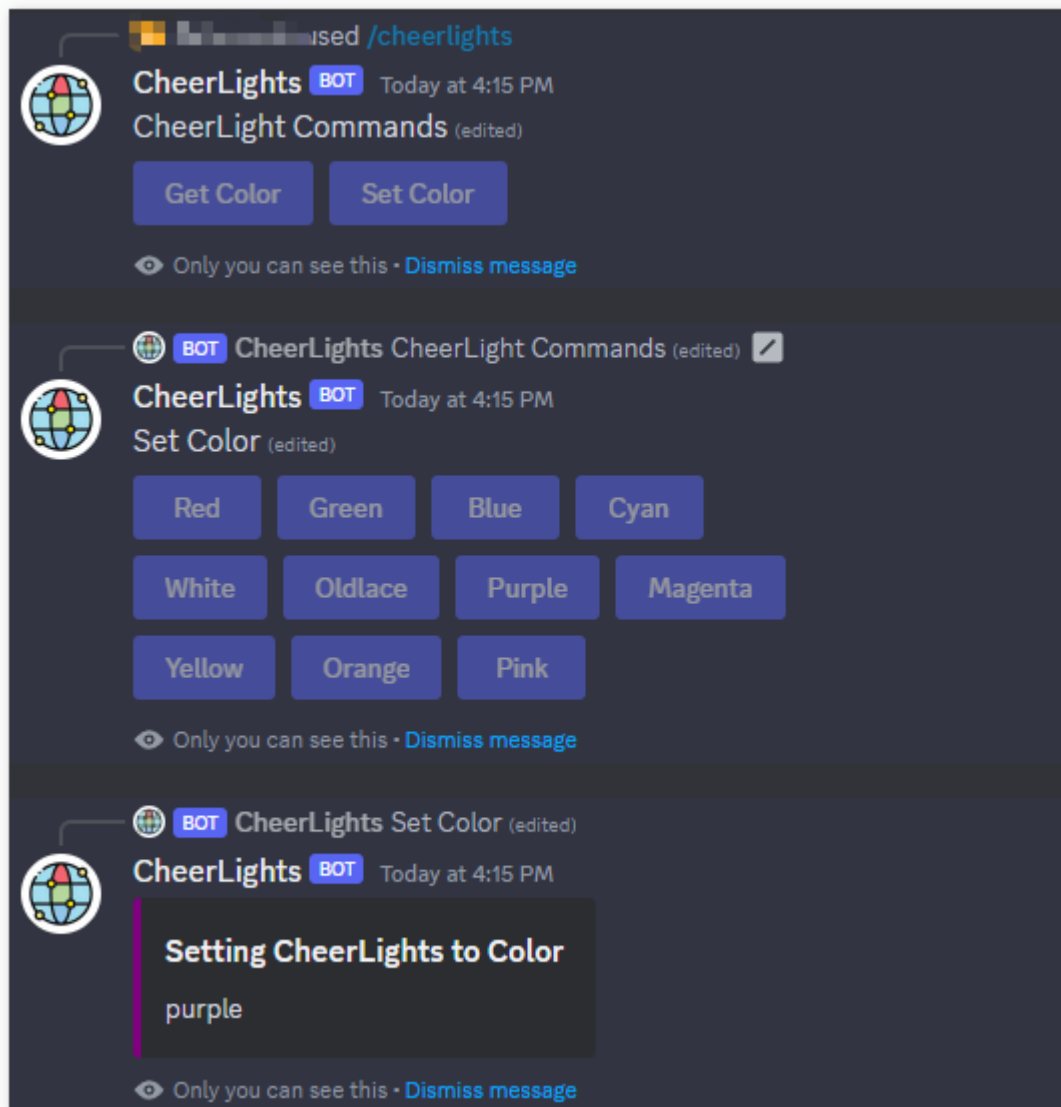
Bemerkung: Im Code werden SSID und Passwort in arduino_secrets.h gespeichert. Bevor Sie dieses Beispiel hochladen, müssen Sie sie mit Ihren eigenen WLAN-Anmeldeinformationen ändern. Außerdem sollten Sie aus Sicherheitsgründen diese Informationen vertraulich halten, wenn Sie den Code teilen oder speichern.

Steuern Sie globale @CheerLights-Geräte

1. Treten Sie dem bei und nutzen Sie den CheerLights-Bot, um die Farbe einzustellen. Geben Sie einfach /cheerlights in einem der Kanäle des **CheerLights Discord Servers** ein, um den Bot zu aktivieren.



2. Folgen Sie den Anweisungen des Bots, um die Farbe festzulegen. Dadurch können Sie CheerLights-Geräte weltweit steuern.



Wie funktioniert des?

Hier sind die Hauptteile des Codes und Erklärungen ihrer Funktionen:

1. Einbinden der erforderlichen Bibliotheken:

- `WiFiS3.h`: Wird verwendet, um Wi-Fi-Verbindungen zu handhaben.
- `ArduinoMqttClient.h`: Wird verwendet, um MQTT-Verbindungen zu handhaben.
- `FastLED.h`: Wird verwendet, um NeoPixel LED-Streifen zu steuern.

2. Definition einiger Konstanten:

- `NUM_LEDS`: Die Anzahl der LEDs auf dem LED-Streifen.
- `DATA_PIN`: Der Datenpin, der mit Arduino verbunden ist, um den LED-Streifen zu steuern.
- `arduino_secrets.h`: Header-Datei, die den Namen und das Passwort des Wi-Fi-Netzwerks enthält, um sensible Informationen zu schützen.
- `broker`: Adresse des MQTT-Servers.

- `port`: Port des MQTT-Servers.
 - `topic`: Das zu abonnierende MQTT-Thema.
3. Definition einiger globaler Variablen:
- `CRGB leds[NUM_LEDS]`: Ein Array, um LED-Farbdaten zu speichern.
 - `colorName`: Ein Array von Farbnamen, die vom CheerLights-Projekt unterstützt werden.
 - `colorRGB`: Ein Array von RGB-Farbcodes, die den Farbnamen entsprechen.
4. `setup()` Funktion:
- Initialisieren der seriellen Kommunikation.
 - Überprüfen, ob das Wi-Fi-Modul vorhanden ist und dessen Firmware-Version ausgeben.
 - Versuchen, eine Verbindung zum Wi-Fi-Netzwerk herzustellen; falls es fehlschlägt, 10 Sekunden warten und erneut versuchen.
 - Nach erfolgreicher Verbindung, Verbindung zum MQTT-Broker (Server) herstellen und das angegebene Thema abonnieren.
 - Den NeoPixel LED-Streifen initialisieren.
5. `loop()` Funktion:
- Periodisch die Funktion `mqttClient.poll()` aufrufen, um MQTT-Nachrichten zu empfangen und MQTT-Keep-Alive-Signale zu senden.
 - Eine 5-Sekunden-Verzögerung hinzufügen, um eine kontinuierliche Verbindung zu vermeiden.
6. Die Funktionen `printWifiData()` und `printCurrentNet()` werden verwendet, um Wi-Fi-Netzwerk- und Verbindungsinformationen auszugeben.
7. Die Funktion `printMacAddress()` wird verwendet, um die MAC-Adresse im Hexadezimalformat auszudrucken.
8. Die Funktion `onMqttMessage()` ist eine Rückruffunktion, die ausgelöst wird, wenn eine MQTT-Nachricht empfangen wird. Sie gibt das empfangene Thema und den Nachrichteninhalte aus und konvertiert den Nachrichteninhalte in Kleinbuchstaben. Wenn das Thema „cheerlights“ ist, ruft sie die Funktion `setColor()` auf, um die Farbe des LED-Streifens festzulegen.
9. Die Funktion `setColor()` nimmt einen Farbnamen als Parameter und sucht nach einer passenden Farbe im Array `colorName`. Wenn eine passende Farbe gefunden wird, setzt sie die Farbe des LED-Streifens auf den entsprechenden RGB-Wert und aktualisiert die Farbe des LED-Streifens mit der Funktion `FastLED.show()`.

7.6 WeatherTime Bildschirm

Dieser Sketch verbindet sich mit einem WLAN-Netzwerk, holt jede Minute Wetterdaten von OpenWeatherMap ab, holt die aktuelle Zeit von einem NTP-Server und zeigt den Tag, die Zeit und die Wetterinformationen auf einem OLED-Bildschirm an.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

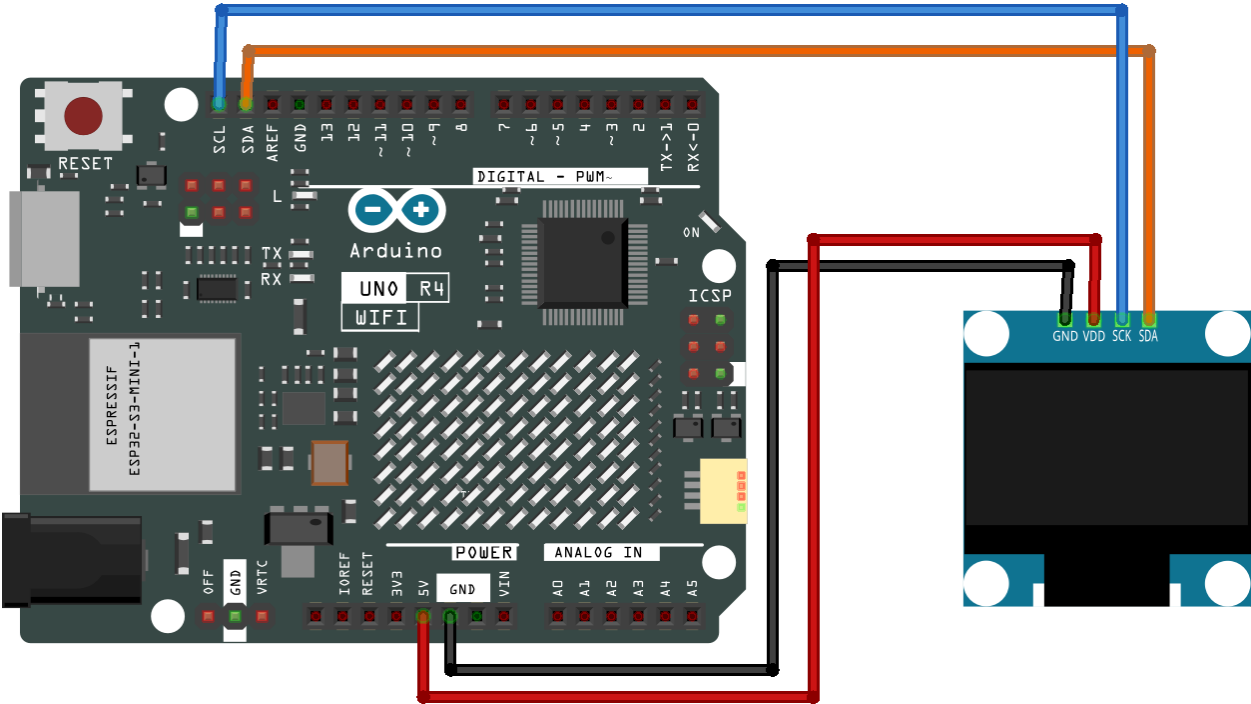
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

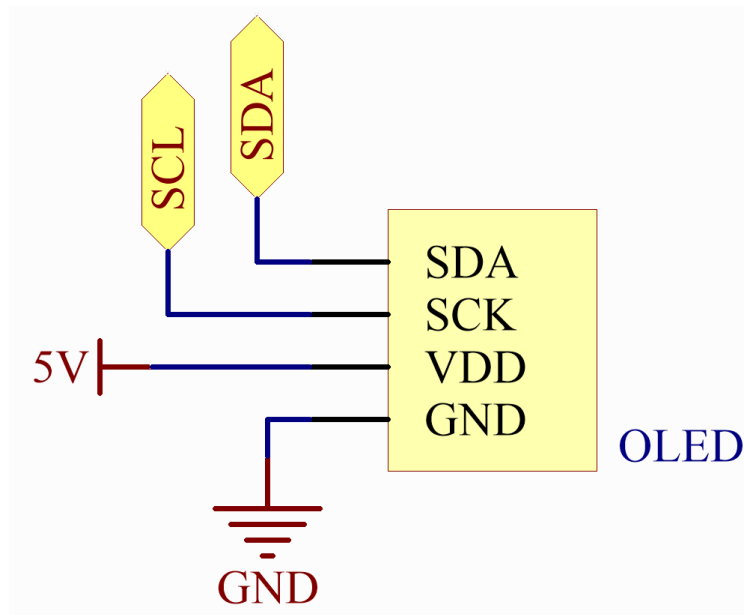
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 Wi-Fi	-
Jumperkabel	
OLED-Display-Modul	

Verdrahtung



Schaltplan



OpenWeather

OpenWeather API-Schlüssel erhalten

OpenWeather ist ein Online-Dienst von OpenWeather Ltd, der weltweite Wetterdaten über eine API bereitstellt, einschließlich aktueller Wetterdaten, Prognosen, Nowcasts und historischer Wetterdaten für jeden geografischen Standort.

1. Besuchen Sie OpenWeather, um sich anzumelden/ein Konto zu erstellen.

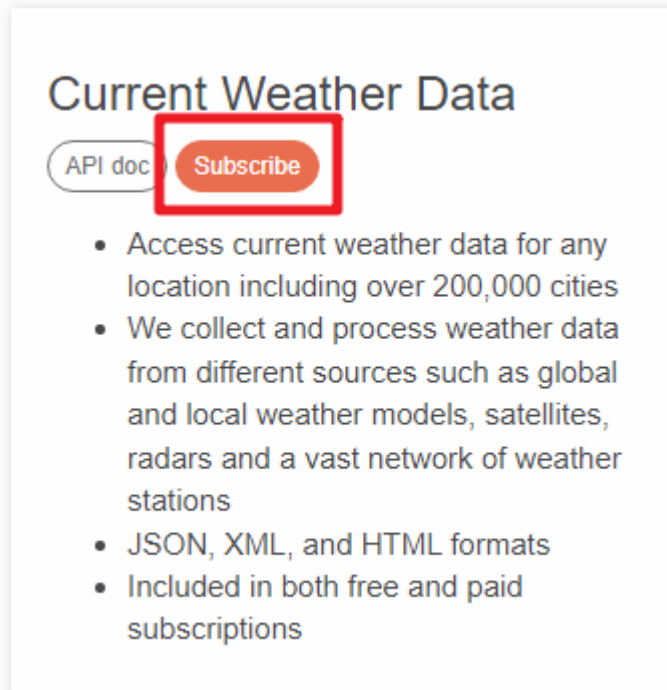
The screenshot shows the 'Sign In To Your Account' form on the OpenWeather website. It includes the following elements:

- Title:** Sign In To Your Account
- Email Field:** A text input field with a user icon and the placeholder text 'Enter email'.
- Password Field:** A text input field with a lock icon and the placeholder text 'Password'.
- Remember me:** A checkbox with the label 'Remember me'.
- Submit Button:** An orange button with the text 'Submit'.
- Links:**
 - 'Not registered? [Create an Account.](#)'
 - 'Lost your password? [Click here to recover.](#)'

2. Klicken Sie in der Navigationsleiste auf die API-Seite.



3. Finden Sie **Current Weather Data** und klicken Sie auf Abonnieren.

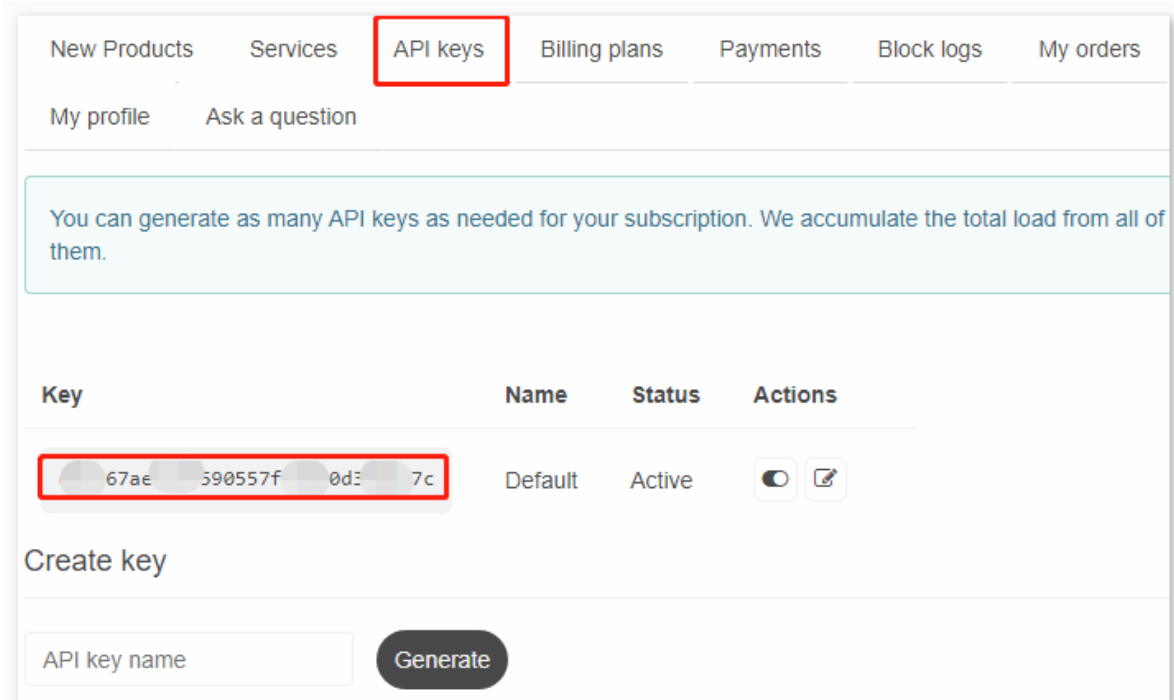


4. Unter **Current weather and forecasts collection** abonnieren Sie den entsprechenden Dienst. In unserem Projekt reicht die kostenlose Version aus.

Current weather and forecasts collection

Free	Startup	Developer	Professional	Enterprise
	40 USD/ month	180 USD/ month	470 USD/ month	from 2000 USD/ month
Get API key	Subscribe	Subscribe	Subscribe	Subscribe
60 calls/minute 1,000,000 calls/month	600 calls/minute 10,000,000 calls/month	3,000 calls/minute 100,000,000 calls/month	30,000 calls/minute 1,000,000,000 calls/month	200,000 calls/minute 5,000,000,000 calls/month
Current Weather	Current Weather	Current Weather	Current Weather	Current Weather
3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days
Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days
Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days
Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days

5. Kopieren Sie den Schlüssel von der Seite **API keys**.

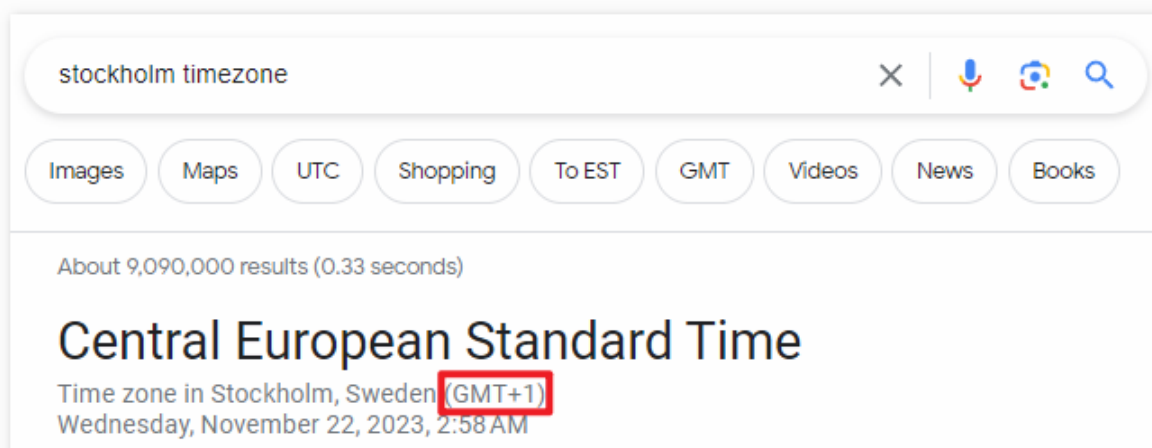


6. Kopieren Sie ihn in die `arduino_secrets.h`.

```
#define SECRET_SSID "<SSID>"           // your network SSID (name)
#define SECRET_PASS "<PASSWORD>"       // your network password
#define API_KEY "<OpenWeather_API_KEY>"
#define LOCATION "<YOUR CITY>"
```

7. Stellen Sie die Zeitzone Ihres Standorts ein.

Nehmen Sie als Beispiel die Hauptstadt Schwedens, Stockholm. Suchen Sie auf Google nach „Stockholm Zeitzone“.



In den Suchergebnissen sehen Sie „GMT+1“, also setzen Sie den Parameter der untenstehenden Funktion auf `3600 * 1` Sekunden.

```
timeClient.setTimeOffset(3600 * 1); // Adjust for your time zone (this is +1 hour)
```

Bibliothek installieren

Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager und suchen Sie nach „ArduinoMqttClient“, „FastLED“, „Adafruit GFX“ und „Adafruit SSD1306“ und installieren Sie diese.

`ArduinoJson.h`: Wird verwendet, um JSON-Daten zu verarbeiten (Daten, die von openweathermap erhalten werden).

`NTPClient.h`: Wird verwendet, um die Echtzeit zu erhalten.

`Adafruit_GFX.h`, `Adafruit_SSD1306.h`: Wird für das OLED-Modul verwendet.

Code ausführen

Bemerkung:

- Sie können die Datei `06_weather_oled.ino` direkt unter dem Pfad `elite-explorer-kit-main\iot_project\06_weather_oled` öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
-

Bemerkung: Im Code werden SSID und Passwort in `arduino_secrets.h` gespeichert. Bevor Sie dieses Beispiel hochladen, müssen Sie sie mit Ihren eigenen WLAN-Anmeldeinformationen ändern. Außerdem sollten Sie aus Sicherheitsgründen diese Informationen vertraulich halten, wenn Sie den Code teilen oder speichern.

Wie funktioniert des?

1. Bibliotheken und Definitionen:

1. **WiFiS3.h**: Dies ist wahrscheinlich eine spezifische Bibliothek für ein bestimmtes WLAN-Modul oder Board zur Verwaltung von WLAN-Verbindungen.
2. **ArduinoJson.h**: Diese Bibliothek wird für das Dekodieren (und Kodieren) von JSON-Daten verwendet.
3. **arduino_secrets.h**: Eine separate Datei, in der sensible Daten (wie WLAN-Zugangsdaten) gespeichert sind. Dies ist eine gute Praxis, um Zugangsdaten aus dem Hauptcode herauszuhalten.
4. **NTPClient & WiFiUdp**: Diese werden verwendet, um die aktuelle Zeit von einem NTP-Server (Network Time Protocol) zu holen.
5. **Adafruit-Bibliotheken**: Verwendet für die Verwaltung des OLED-Displays.
6. **Verschiedene globale Variablen**: Dazu gehören WLAN-Zugangsdaten, Serverdetails und mehr, die im gesamten Skript verwendet werden.

2. `setup()`:

1. Initialisiert die serielle Kommunikation.
2. Überprüft und druckt die Firmware-Version des WLAN-Moduls aus.
3. Versucht, eine Verbindung zum WLAN-Netzwerk mit der bereitgestellten SSID und dem Passwort herzustellen.
4. Gibt den Status des verbundenen WLANs aus (SSID, IP, Signalstärke).
5. Initialisiert das OLED-Display.
6. Startet den NTP-Client, um die aktuelle Zeit zu holen und legt einen Zeitversatz fest (in diesem Fall 8 Stunden, was einer bestimmten Zeitzone entsprechen könnte).

3. `read_response()`:

1. Liest die Antwort vom Server, insbesondere auf der Suche nach JSON-Daten (gekennzeichnet durch { und }).
 2. Wenn JSON-Daten gefunden werden, werden die Daten dekodiert, um Wetterdetails wie Temperatur, Feuchtigkeit, Druck, Windgeschwindigkeit und Windrichtung zu extrahieren.
 3. Ruft die Funktion `displayWeatherData` auf, um die Wetterinformationen auf dem OLED-Bildschirm anzuzeigen.
4. `httpRequest()`:
1. Schließt alle bestehenden Verbindungen, um sicherzustellen, dass der Socket des WLAN-Moduls frei ist.
 2. Versucht, eine Verbindung zum OpenWeatherMap-Server herzustellen.
 3. Wenn verbunden, sendet es eine HTTP-GET-Anfrage, um die Wetterdaten für einen spezifischen Ort zu holen, der durch `LOCATION` definiert wird (wahrscheinlich definiert in `arduino_secrets.h` oder anderswo).
 4. Zeichnet die Zeit auf, zu der die Anfrage gestellt wurde.
5. `loop()`:
1. Ruft die Funktion `read_response` auf, um alle eingehenden Daten vom Server zu verarbeiten.
 2. Aktualisiert die Zeit vom NTP-Server.
 3. Überprüft, ob es Zeit ist, eine weitere Anfrage an den Wetter-Server zu stellen (basierend auf dem `postingInterval`). Wenn ja, ruft es die Funktion `httpRequest` auf.
6. `printWifiStatus()`:
1. Die SSID des verbundenen Netzwerks.
 2. Die lokale IP-Adresse des Boards.
 3. Die Signalstärke (RSSI).
7. `displayWeatherData()`:
1. Löscht den OLED-Bildschirm.
 2. Zeigt den aktuellen Wochentag an.
 3. Zeigt die aktuelle Uhrzeit im HH:MM-Format an.
 4. Zeigt die bereitgestellten Wetterdaten an (Temperatur, Feuchtigkeit, Druck und Windgeschwindigkeit).

7.7 Bluetooth-Nachrichtenbox

Dieses Projekt empfängt Nachrichten und zeigt sie auf einem LCD-Bildschirm an.

Sie können es als Familien-Nachrichtentafel nutzen, um Familienmitglieder, die noch nicht gegangen sind, daran zu erinnern, ihre Schlüssel mitzunehmen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

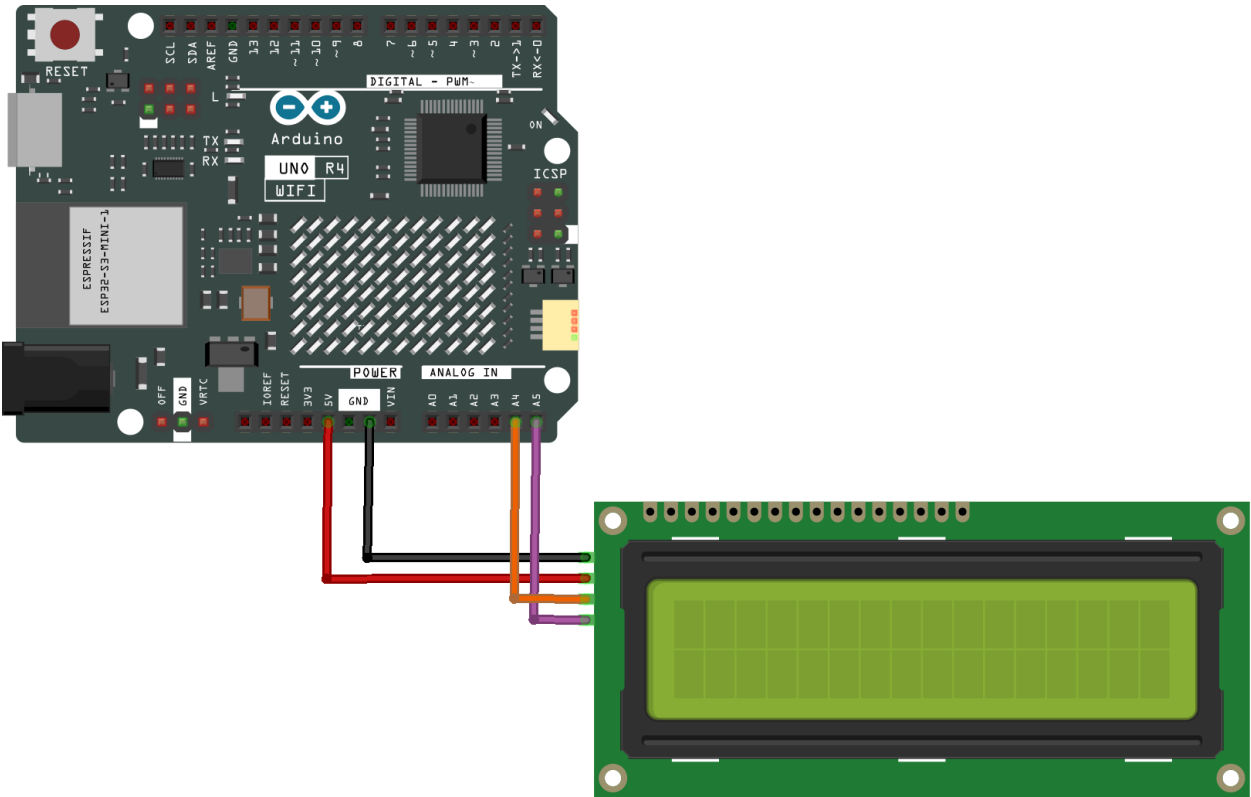
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

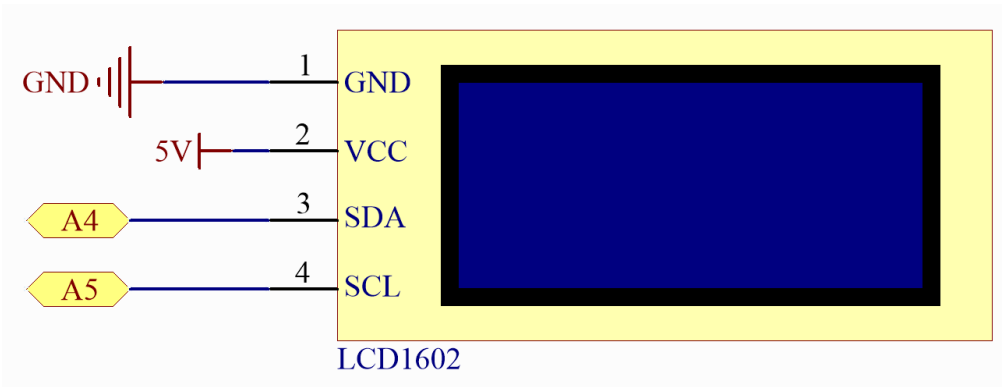
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Steckbrett	
Jumperkabel	
I2C LCD1602	

Verdrahtung



Schaltplan



UUID generieren

UUIDs spielen eine entscheidende Rolle bei der BLE-Kommunikation und gewährleisten die Einzigartigkeit von Geräten sowie die Genauigkeit des Datenaustauschs zwischen ihnen. Sie müssen UUIDs anpassen, um Ihre eigenen BLE-Dienste und -Merkmale für spezifische Anforderungen zu erstellen. (Hier müssen wir ein Merkmal erstellen, das Texteingaben unterstützt.)

1. Verwenden Sie das [Online UUID Generator Tool](#), um UUIDs zu erstellen, die einzigartig für Sie sind, um UUID-Konflikte zu vermeiden.
2. Erzeugen Sie zwei UUIDs der Version 4.

3. Kopieren Sie sie und ersetzen Sie die beiden UUIDs in Ihrem Code.

```
#define SERVICE_UUID "uuid1"
#define CHARACTERISTIC_UUID "uuid2"
```

Bibliothek installieren

ArduinoBLE.h: Wird für die Verwaltung von Bluetooth Low Energy (BLE) Kommunikation verwendet.
LiquidCrystal_I2C.h: Wird verwendet, um einen 16x2-Zeichen-LCD-Bildschirm mit einer I2C-Schnittstelle zu steuern.

Code ausführen

Bemerkung:

- Sie können die Datei 07_lightblue_lcd.ino direkt unter dem Pfad elite-explorer-kit-main\iot_project\07_lightblue_lcd öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager, um nach „Ardui-

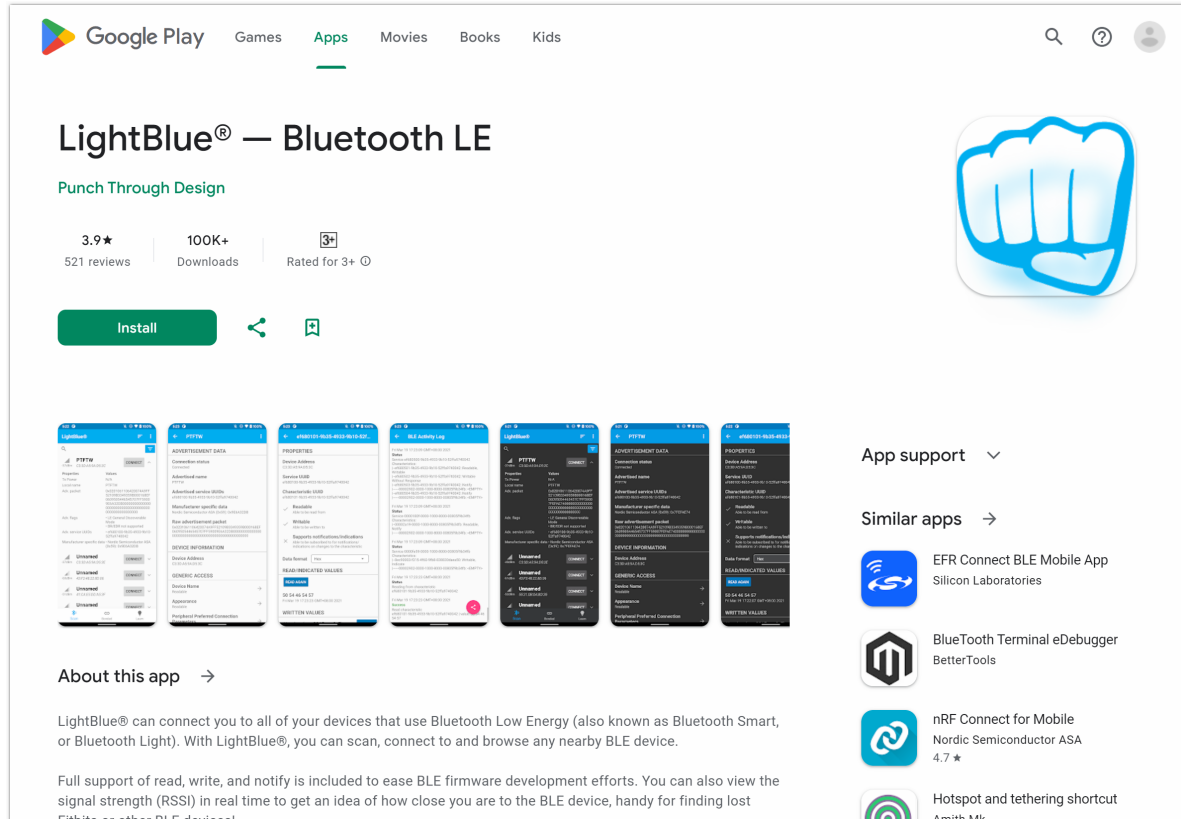
noBLE“ und „LiquidCrystal I2C“ zu suchen und diese zu installieren.

Wie funktioniert des?

Um mit den in diesem Sketch erstellten Diensten und Merkmalen zu interagieren, sollten Sie eine generische Bluetooth® Low Energy-Zentral-App wie LightBlue (verfügbar für iOS und Android) oder nRF Connect (für Android) verwenden.

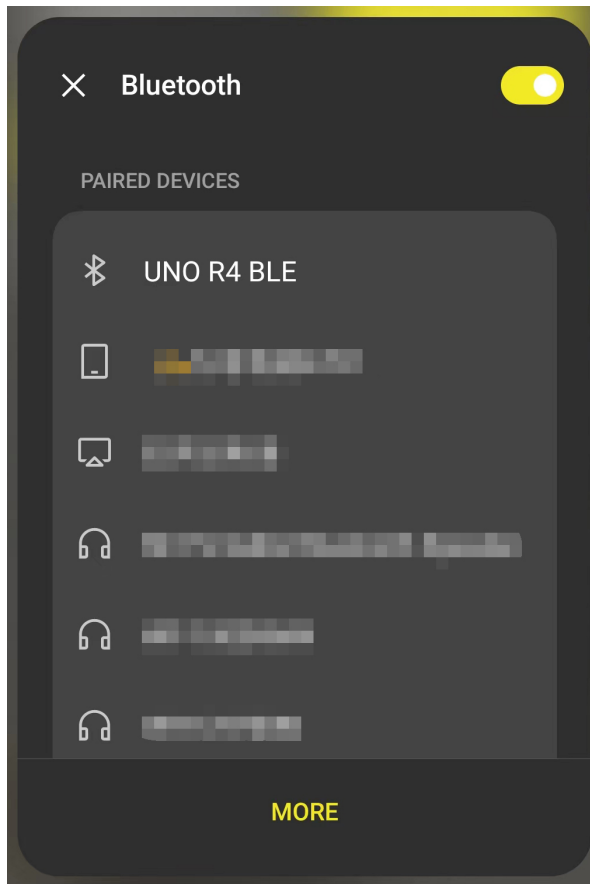
Wir verwenden LightBlue als Beispiel, um zu demonstrieren, wie man Arduino's LED über Bluetooth steuert.

1. Laden Sie die LightBlue-App aus dem App Store (für iOS) oder Google Play (für Android) herunter.



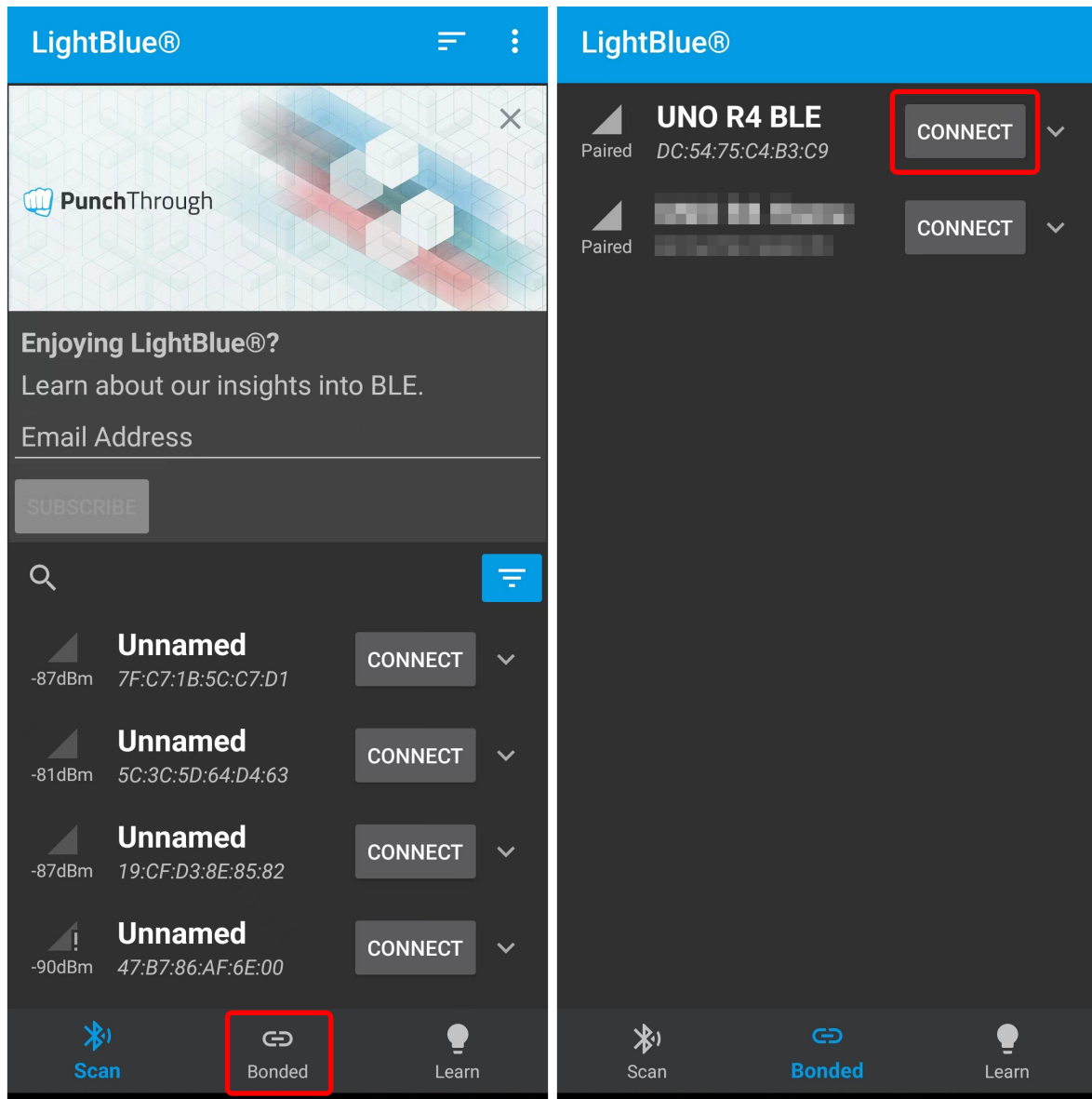
2. Verbinden des Arduino mit Ihrem Smartphone über Bluetooth

Gehen Sie zu Ihren Bluetooth-Einstellungen und suchen Sie das Gerät namens „UNO R4 BLE.“ Stellen Sie eine Verbindung dazu her.



3. Interaktion mit Arduino über Bluetooth mit LightBlue

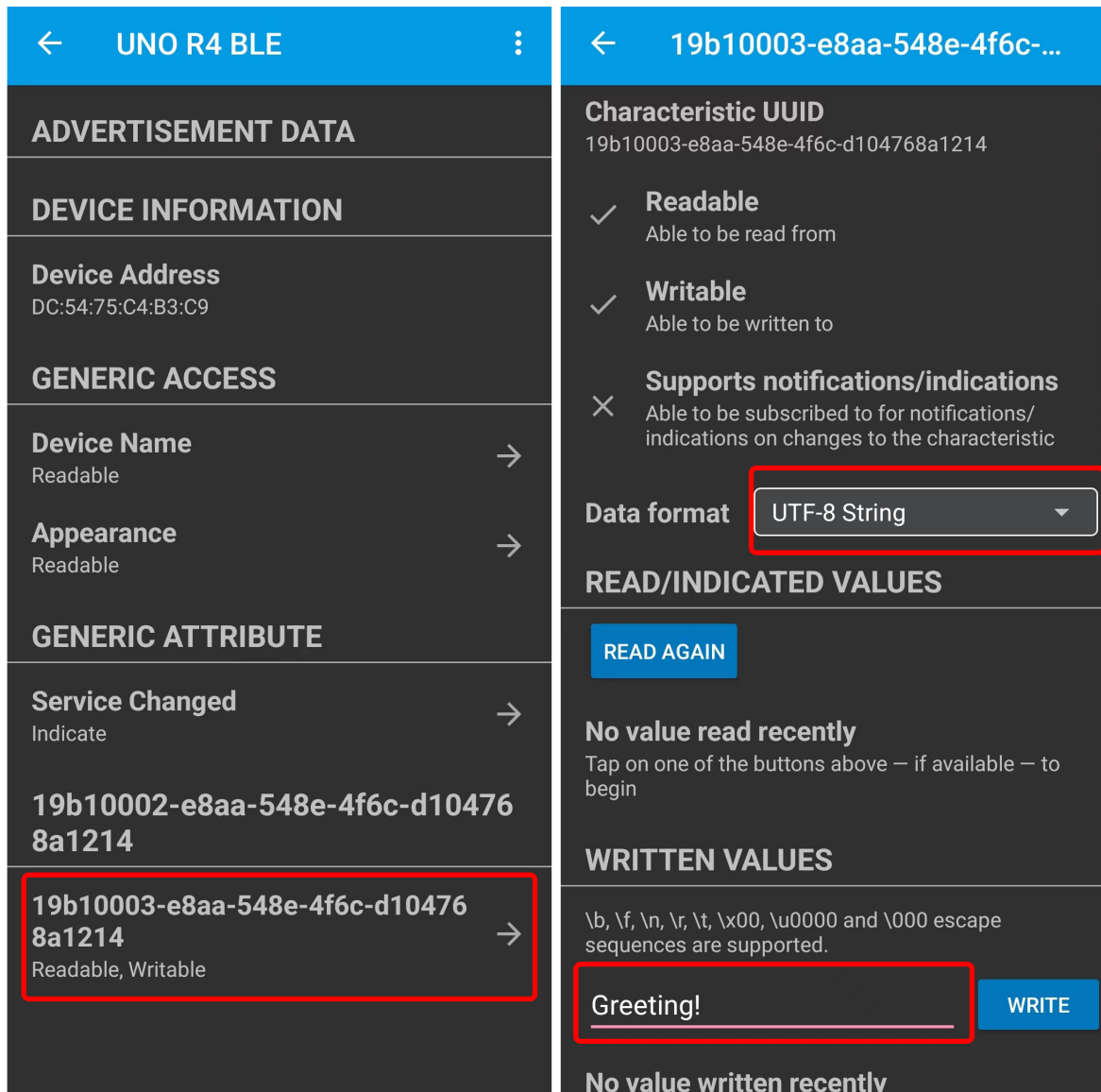
Starten Sie LightBlue und tippen Sie auf den Reiter **Bonded** unten in der Schnittstelle. Hier sehen Sie eine Liste der BLE-Geräte, mit denen Ihr Smartphone zuvor gekoppelt wurde. Finden Sie **UNO R4 BLE** und tippen Sie auf **CONNECT**.



Nachdem Sie verbunden sind, erhalten Sie Zugang zu detaillierten Informationen über das Bluetooth-Gerät. Scrollen Sie nach unten, um Ihre Dienst-UUID und Ihre Charakteristik-UUID zu finden.

Tippen Sie auf die Charakteristik. Sie werden feststellen, dass diese Charakteristik sowohl lesbar als auch beschreibbar ist, was Ihnen erlaubt, sowohl davon zu lesen als auch darauf zu schreiben.

Scrollen Sie weiter zum Abschnitt „Datenformat“ und ändern Sie ihn auf UTF-8-String. Geben Sie Text in das Textfeld ein und klicken Sie auf schreiben. Die von Ihnen eingegebene Nachricht erscheint auf dem LCD.



Wie funktioniert des?

Hier sind die Hauptteile des Codes und Erklärungen ihrer Funktionen:

1. Einbinden der erforderlichen Bibliotheken:

- **ArduinoBLE.h:** Wird für die Handhabung von BLE-Kommunikation verwendet.
- **Wire.h:** Wird für I2C-Kommunikation verwendet.
- **LiquidCrystal_I2C.h:** Wird verwendet, um einen 16x2-Zeichen-LCD-Bildschirm mit einer I2C-Schnittstelle zu steuern.

2. Definieren eines BLE-Dienstes und einer BLE-Charakteristik:

- Verwenden Sie die Klasse **BLEService**, um einen BLE-Dienst zu definieren und ihm eine einzigartige UUID zuzuweisen.
- Verwenden Sie die Klasse **BLECharacteristic**, um eine BLE-Charakteristik zu definieren, ihr eine einzigartige UUID zuzuweisen und ihr Lese- (BLERead) und Schreibrechte (BLEWrite) zu geben.
- Erstellen Sie ein Zeichen-Array **stringValue** als Anfangswert der Charakteristik.

3. Initialisieren eines 16x2-Zeichen-LCD-Bildschirms (LCD):

- Verwenden Sie die Klasse `LiquidCrystal_I2C`, um ein LCD zu initialisieren, indem Sie die I2C-Adresse (0x27) und die Anzahl der Zeilen und Spalten (16x2) angeben.
- Schalten Sie die Hintergrundbeleuchtung des LCD ein, löschen Sie den Bildschirm, bewegen Sie den Cursor zum Anfang der ersten Zeile und zeigen Sie „Bluetooth LCD“ an.

4. Durchführung der Initialisierung in der Funktion `setup()`:

- Initialisieren der seriellen Kommunikation.
- Initialisieren des BLE-Moduls und, falls die Initialisierung fehlschlägt, in eine unendliche Schleife eintreten.
- Setzen des lokalen Namens und der Service-UUID für das BLE-Peripheriegerät.
- Hinzufügen der BLE-Charakteristik zum BLE-Dienst.
- Starten der Werbung für den BLE-Dienst, damit zentrale Geräte ihn entdecken und sich damit verbinden können.
- Initialisieren des LCD.

5. Die Funktion `loop()`:

- Überprüfen, ob ein zentrales Gerät mit dem BLE-Peripheriegerät verbunden ist, indem `BLE.central()` aufgerufen wird. Wenn ein zentrales Gerät verbunden ist, in die Logik der Verbindungsbearbeitung eintreten.
- Im verbundenen Zustand überprüfen, ob Daten in die BLE-Charakteristik geschrieben wurden, indem `boxCharacteristic.written()` überprüft wird.
- Wenn Daten geschrieben wurden, die Länge der geschriebenen Daten mit `boxCharacteristic.valueLength()` ermitteln und ein Byte-Array `buffer` erstellen, um die geschriebenen Daten zu speichern.
- Verwenden Sie `boxCharacteristic.readValue()`, um Daten aus der BLE-Charakteristik in den `buffer` zu lesen.
- Ein Nullzeichen `'\0'` an das Ende des `buffer` hinzufügen, um es in einen String zu konvertieren.
- Die empfangene Nachricht auf dem seriellen Monitor ausgeben und auf dem LCD anzeigen.
- Weiterhin auf das nächste zentrale Gerät warten, das sich verbindet, nachdem das zentrale Gerät getrennt wurde.

7.8 Bluetooth-Umweltmonitor

Dieses Projekt verwendet eine Android-App, die mit MIT App Inventor erstellt wurde, um Umweltdaten von einem Arduino-Board zu empfangen und anzuzeigen. Das Arduino-Board holt Daten von einem DHT11-Sensor, um Temperatur und Feuchtigkeit zu messen. Sobald die Daten gesammelt sind, werden sie über Bluetooth übertragen. Die App zeigt die Daten auf dem Bildschirm an, sobald sie empfangen werden.

Die Android-Anwendung wird unter Verwendung einer kostenlosen webbasierten Plattform namens `MIT App Inventor` erstellt. Das Projekt bietet eine ausgezeichnete Gelegenheit, sich mit der Schnittstelle zwischen einem Arduino und einem Smartphone vertraut zu machen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

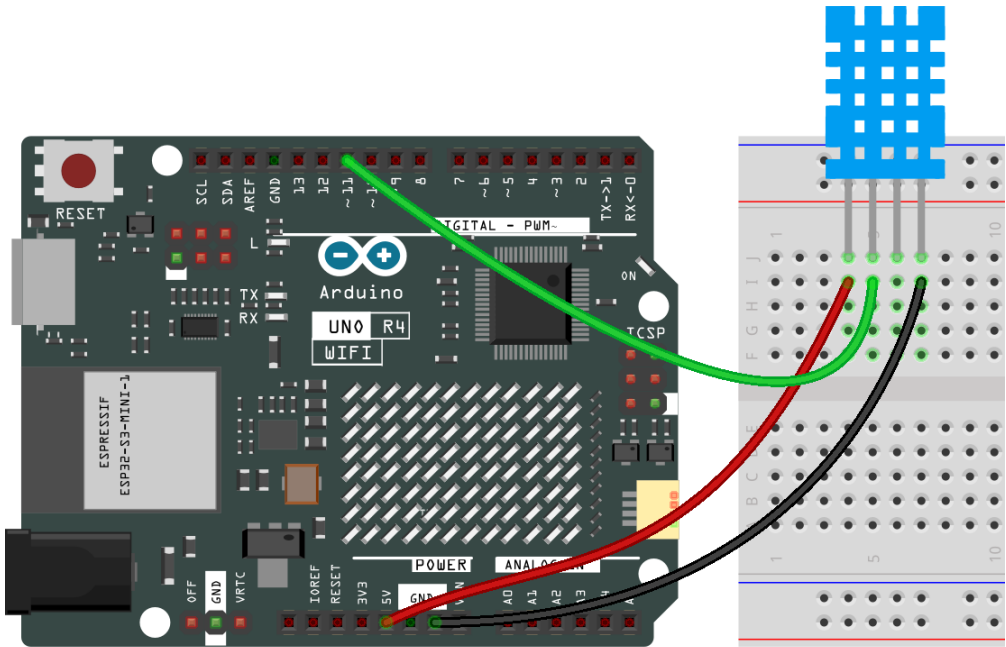
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

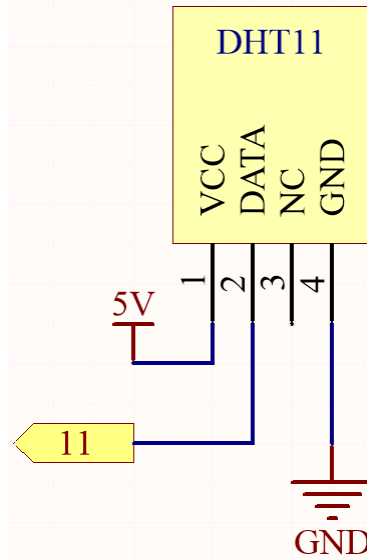
Name	ARTIKEL IN DIESEM KIT	LINK
Elite Explorer Kit	300+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 WiFi	-
Steckbrett	
Jumperkabel	
Feuchtigkeitssensor-Modul	

1. Bauen Sie den Schaltkreis



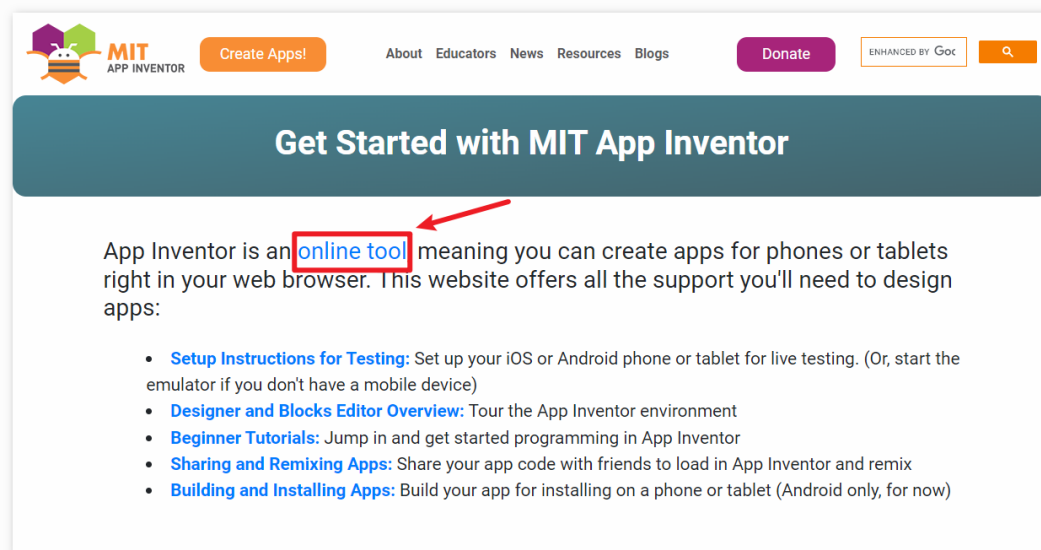


2. Erstellen Sie die Android-App

Die Android-Anwendung wird unter Verwendung einer kostenlosen Webanwendung namens entwickelt. MIT App Inventor dient als ausgezeichnete Einstiegspunkt für die Android-Entwicklung, dank seiner intuitiven Drag-and-Drop-Funktionen, die die Erstellung einfacher Anwendungen ermöglichen.

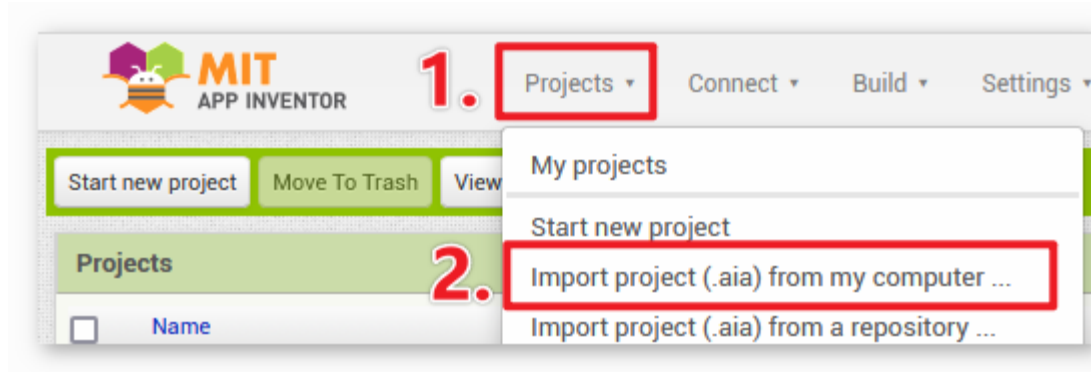
Lassen Sie uns beginnen.

1. Gehen Sie zu und klicken Sie auf „Online-Tool“, um sich anzumelden. Sie benötigen ein Google-Konto, um sich bei MIT App Inventor zu registrieren.

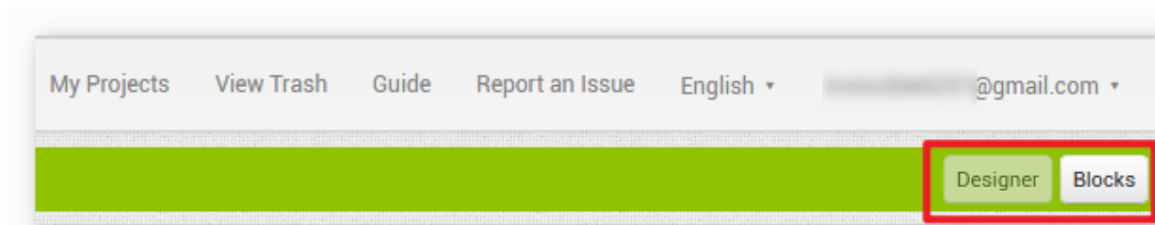


2. Nach dem Einloggen navigieren Sie zu **Projects -> Import project (.aia) from my computer**. Laden Sie anschließend die Datei `ble_environmental_monitor.aia` hoch, die sich im Pfad `elite-explorer-kit-main\iot_project\08-bluetooth_environmental_monitor` befindet.

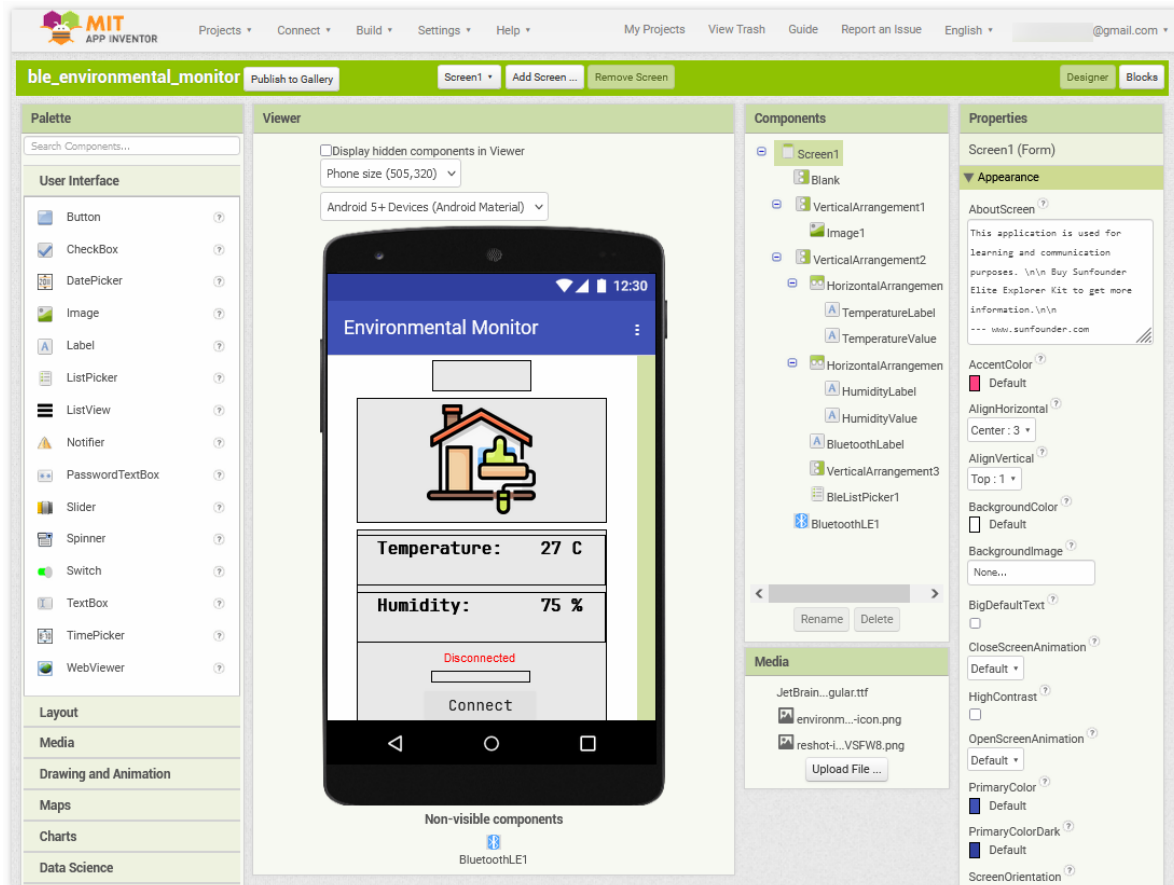
Sie können sie auch direkt hier herunterladen: `ble_environmental_monitor.aia`



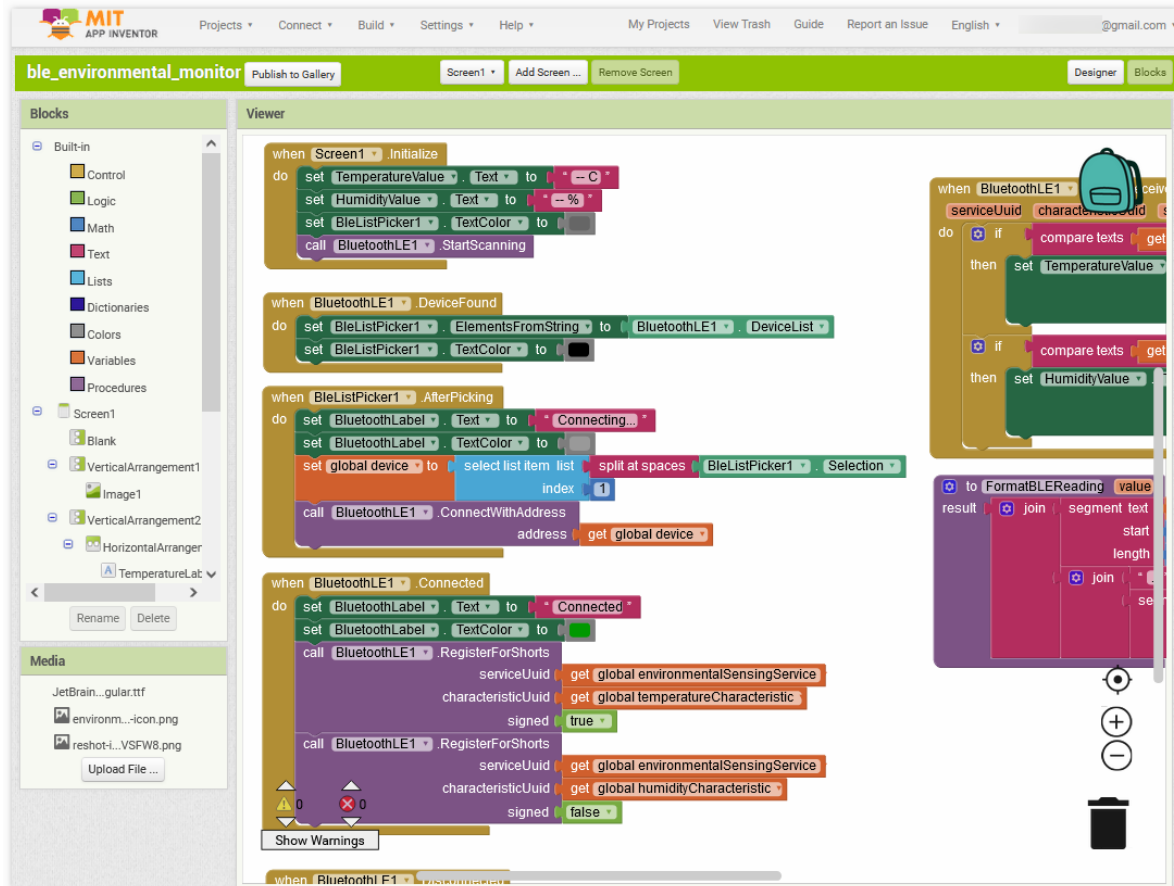
3. Nach dem Hochladen der .aia-Datei sehen Sie die Anwendung in der MIT App Inventor-Software. Dies ist eine vorkonfigurierte Vorlage. Sie können diese Vorlage ändern, nachdem Sie sich mit MIT App Inventor vertraut gemacht haben, indem Sie die folgenden Schritte befolgen.
4. In MIT App Inventor haben Sie 2 Hauptbereiche: den **Designer** und die **Blocks**. Sie können zwischen diesen beiden Bereichen in der oberen rechten Ecke der Seite wechseln.



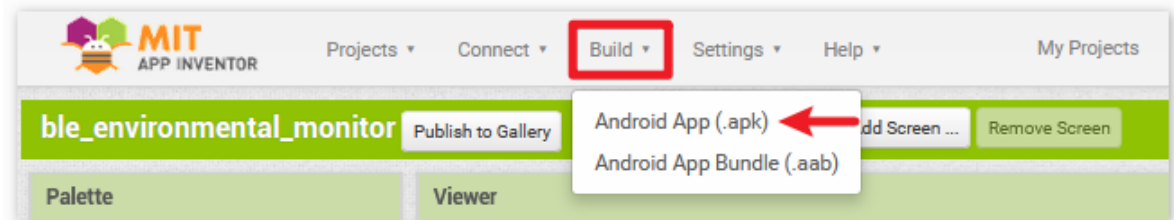
5. Der **Designer** ermöglicht es Ihnen, Buttons, Text, Bildschirme hinzuzufügen und das Gesamtaussehen Ihrer Anwendung zu ändern.



6. Als Nächstes gibt es den Bereich **Blocks**. Dieser Abschnitt ermöglicht es Ihnen, benutzerdefinierte Funktionalitäten für Ihre App zu erstellen, indem Sie jedes Element in der grafischen Benutzeroberfläche der App programmieren, um die gewünschten Funktionen zu erreichen.



7. Um die Anwendung auf einem Smartphone zu installieren, navigieren Sie zum Reiter **Build**.



- Sie können eine `.apk`-Datei erstellen. Nachdem Sie diese Option ausgewählt haben, erscheint eine Seite, auf der Sie zwischen dem Herunterladen einer `.apk`-Datei oder dem Scannen eines QR-Codes für die Installation wählen können. Folgen Sie der Installationsanleitung, um die Installation der Anwendung abzuschließen.

Sie können auch unsere vorab kompilierte APK hier herunterladen: `ble_environmental_monitor.apk`

- Wenn Sie diese App auf Google Play oder einem anderen App-Marktplatz hochladen möchten, können Sie eine `.aab`-Datei generieren.

3. Code hochladen

1. Öffnen Sie die Datei `08-bluetooth_environmental_monitor.ino` unter dem Pfad `elite-explorer-kit-main\iot_project\08-bluetooth_environmental_monitor`, oder kopieren Sie diesen Code in die **Arduino IDE**.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager, um nach

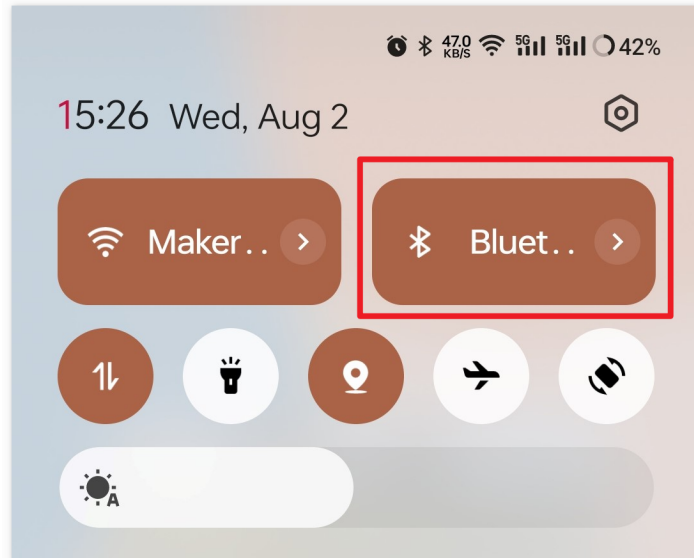
„DHT sensor library“ und „ArduinoBLE“ zu suchen und diese zu installieren.

2. Nachdem Sie das richtige Board und den richtigen Port ausgewählt haben, klicken Sie auf die Schaltfläche **Hochladen**.
3. Öffnen Sie den Seriellen Monitor (Baudrate auf **9600** einstellen), um Debug-Nachrichten zu sehen.

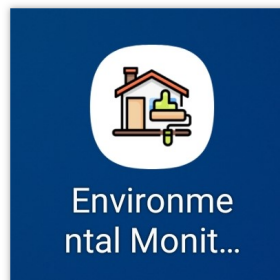
4. App- und Bluetooth-Modul-Verbindung

Stellen Sie sicher, dass die zuvor erstellte Anwendung auf Ihrem Smartphone installiert ist.

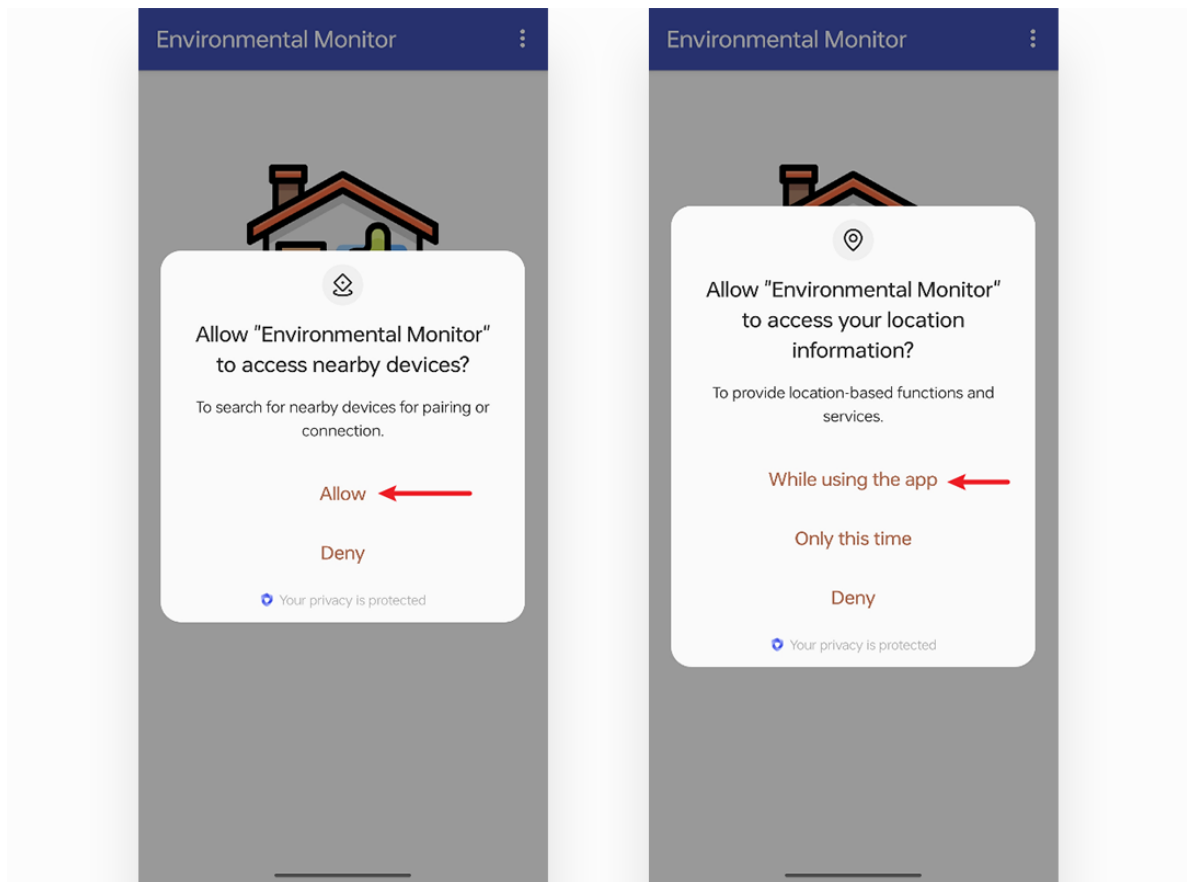
1. Schalten Sie zunächst das **Bluetooth** auf Ihrem Smartphone ein.



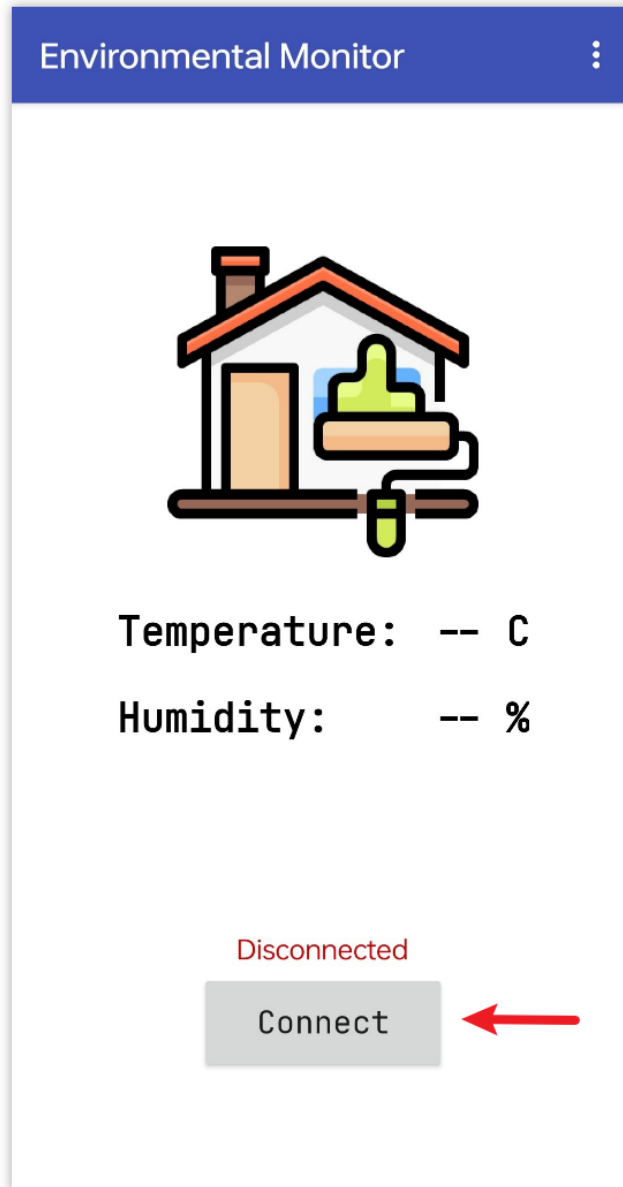
2. Öffnen Sie nun die neu installierte **Environmental Monitor-APP**.



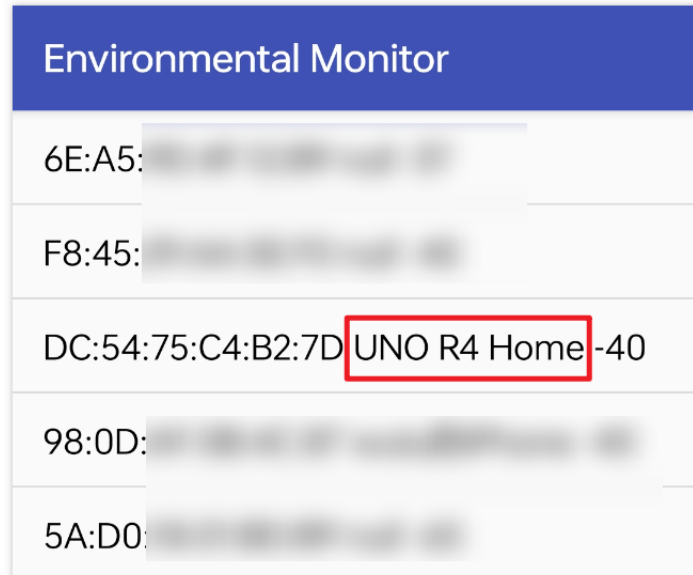
3. Wenn Sie diese App zum ersten Mal öffnen, erscheinen nacheinander zwei Autorisierungsaufforderungen. Dies sind die Berechtigungen, die für die Verwendung von Bluetooth erforderlich sind.



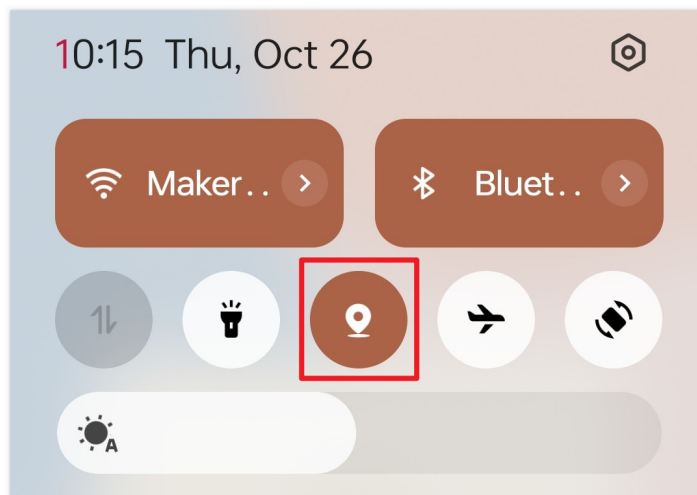
4. In der App klicken Sie auf die Schaltfläche **Connect**, um eine Verbindung zwischen der App und dem Bluetooth-Modul herzustellen.



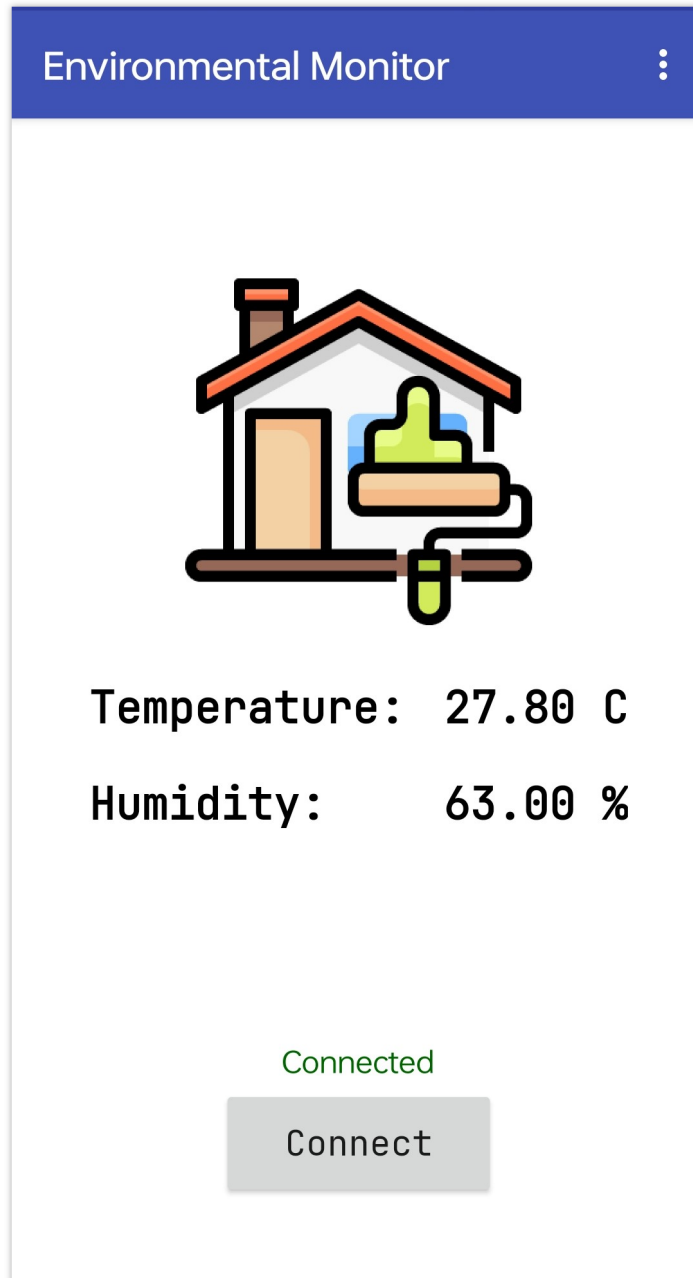
5. Diese Seite zeigt eine Liste aller gekoppelten Bluetooth-Geräte an. Wählen Sie die Option `xx.xx.xx.xx.xx.xx UNO R4 Home` aus der Liste. Der Name jedes Geräts ist neben seiner MAC-Adresse aufgeführt.



6. Wenn Sie auf der oben gezeigten Seite keine Geräte sehen, können Sie versuchen, den Standort-Schalter des Geräts einzuschalten (einige Android-Systemversionen bündeln den Standort-Schalter mit der Bluetooth-Funktion).



7. Nach einer erfolgreichen Verbindung werden Sie zur Hauptseite umgeleitet, auf der Temperatur und Luftfeuchtigkeit angezeigt werden.



5. Code-Erklärung

1. Importieren von Bibliotheken und Definieren von Konstanten

- Importieren Sie die erforderlichen Bibliotheken und definieren Sie Konstanten für den DHT-Sensor-Pin und -Typ.

Bemerkung: Um die Bibliothek zu installieren, verwenden Sie den Arduino-Bibliotheksmanager, um nach „DHT sensor library“ und „ArduinoBLE“ zu suchen und diese zu installieren.

```
#include <DHT.h>
#include <ArduinoBLE.h>
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#define DHTPIN 11
#define DHTTYPE DHT11
```

2. Initialisierung von BLE-Diensten und -Merkmale

- Definieren Sie die UUIDs für BLE Environmental Sensing Service und Merkmale. Wir verwenden die von bereitgestellten vordefinierten UUIDs. Der wird 0x181A zugewiesen, während 0x2A6E und 0x2A6F für reserviert sind.

```
BLEService environmentalSensingService("181A");
BLEShortCharacteristic temperatureCharacteristic("2A6E", BLERead | BLENotify);
BLEUnsignedShortCharacteristic humidityCharacteristic("2A6F", BLERead | BLENotify);
```

3. Setup-Funktion

- Initialisieren Sie die serielle Kommunikation, den DHT-Sensor und BLE.

```
void setup() {
  Serial.begin(9600);
  dht.begin();
  if (!BLE.begin()) {
    Serial.println("starting Bluetooth® Low Energy module failed!");
    while (1)
      ;
  }
  setupBle();
}
```

4. Haupt-Schleife

- Abfragen von BLE-Ereignissen und regelmäßiges Aktualisieren der Sensordaten.
- Die Zeile `millis() - lastUpdateTime > updateInterval` stellt sicher, dass die Sensordaten alle `updateInterval` Millisekunden aktualisiert werden.

```
void loop() {
  BLE.poll();
  if (millis() - lastUpdateTime > updateInterval) {
    // Read sensor data and update BLE characteristics
  }
}
```

5. BLE- und Debug-Funktionen

Funktionen für das Einrichten von BLE, das Drucken von Debug-Informationen und das Verwalten von BLE-Ereignissen.

```
void printDHT(float h, float t) { /* ... */ }
void setupBle() { /* ... */ }
void blePeripheralConnectHandler(BLEDevice central) { /* ... */ }
void blePeripheralDisconnectHandler(BLEDevice central) { /* ... */ }
```

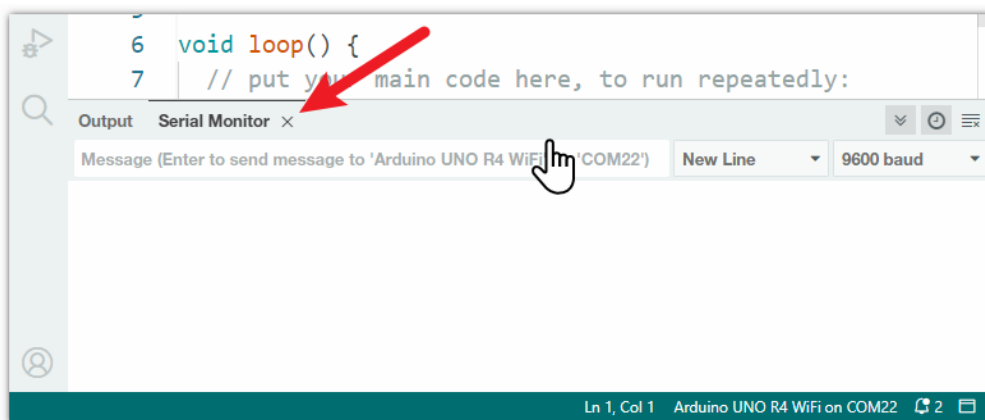
- `printDHT(float h, float t)`: Wird verwendet, um die vom DHT11 gelesenen Temperatur- und Feuchtigkeitsdaten auf dem seriellen Monitor auszudrucken. Diese Funktion dient zu Debugging-Zwecken.
- `setupBle()`: Initialisiert Bluetooth, einschließlich der Einstellung des Sendenamens, der Merkmale und Dienste.

- `blePeripheralConnectHandler(BLEDevice central)` und `blePeripheralDisconnectHandler(BLEDevice central)`: Diese Funktionen behandeln Ereignisse, wenn Bluetooth sich verbindet oder trennt. Wenn ein Gerät erfolgreich über Bluetooth mit dem UNO R4 verbunden ist, leuchtet die LED an Bord. Wenn das Gerät die Verbindung trennt, schaltet sich die LED aus.

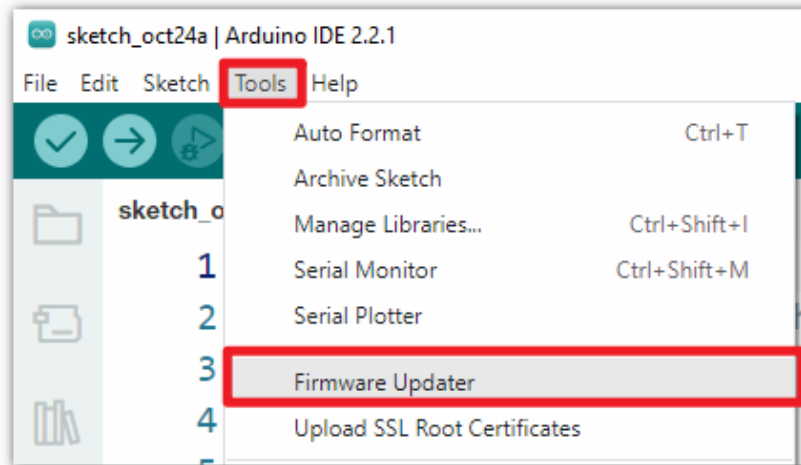
8.1 Aktualisieren der Radio-Modul-Firmware auf Ihrem UNO R4 WiFi-Board

1. Verbinden Sie das UNO R4 WiFi-Board mit Ihrem Computer über das USB-Kabel.
2. Wenn der Arduino IDE Serielle Monitor oder der Serielle Plotter geöffnet sind, schließen Sie diese.

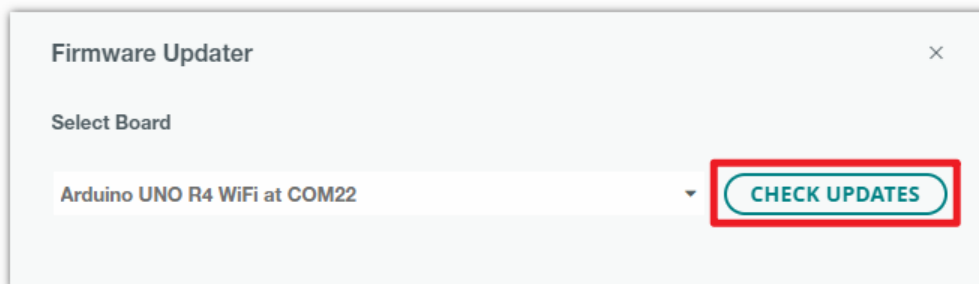
Sie können den Seriellen Monitor schließen, indem Sie auf das X-Symbol klicken, das auf seinem Tab erscheint, wenn es ausgewählt ist:



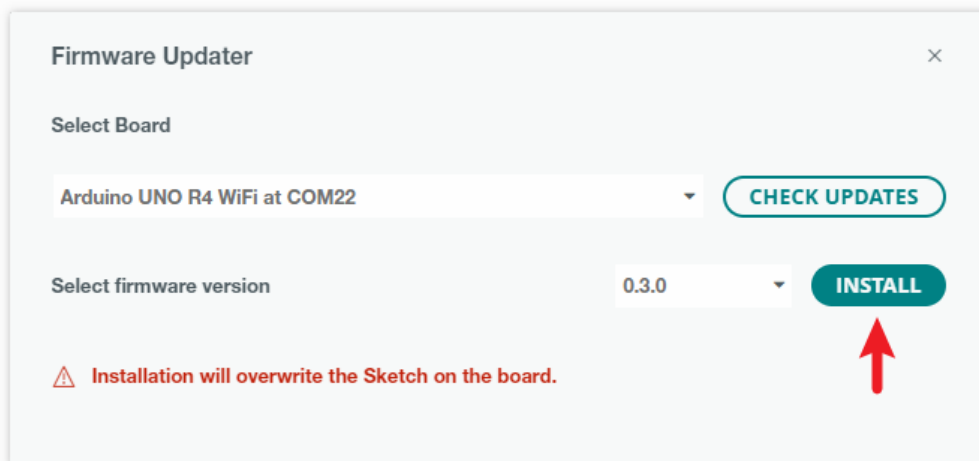
3. Wählen Sie Werkzeuge > Firmware Updater in den Arduino IDE-Menüs aus. Das Dialogfenster „Firmware Updater“ wird geöffnet.



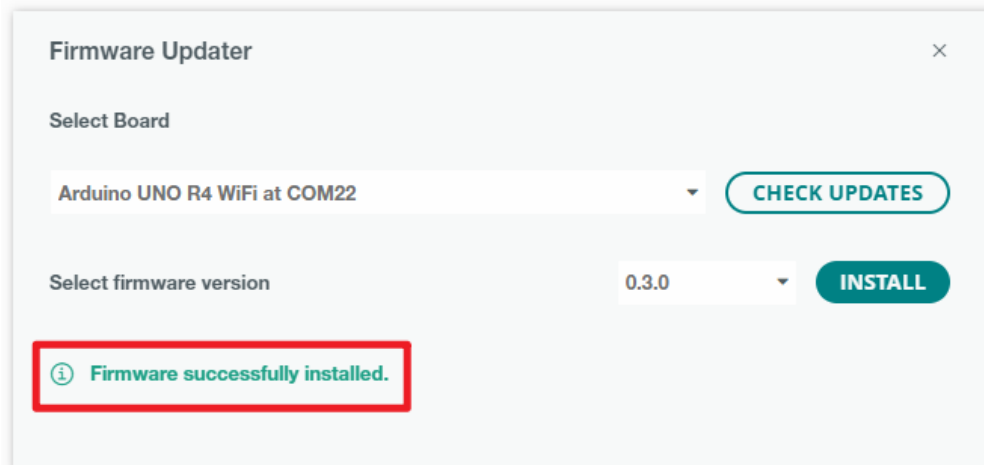
4. Wählen Sie das UNO R4 WiFi aus dem Menü „Board auswählen“ im Dialog „Firmware Updater“.
5. Klicken Sie auf die Schaltfläche „CHECK UPDATES“. Eine Schaltfläche „INSTALL“ wird zum Dialog hinzugefügt.



6. Klicken Sie auf die Schaltfläche „INSTALL“. Ein Prozess „Firmware installieren“ beginnt, wie durch die Nachricht am unteren Rand des Dialogs angezeigt.



7. Warten Sie, bis der Firmware-Update-Prozess erfolgreich abgeschlossen ist, wie durch die Nachricht im Dialog angezeigt: Firmware erfolgreich installiert.



8. Klicken Sie auf das X-Symbol im Dialog. Das Dialogfenster wird geschlossen.
9. Trennen Sie das USB-Kabel des UNO R4 WiFi-Boards von Ihrem Computer.
10. Verbinden Sie das UNO R4 WiFi-Board erneut mit Ihrem Computer über das USB-Kabel.

Referenz

-

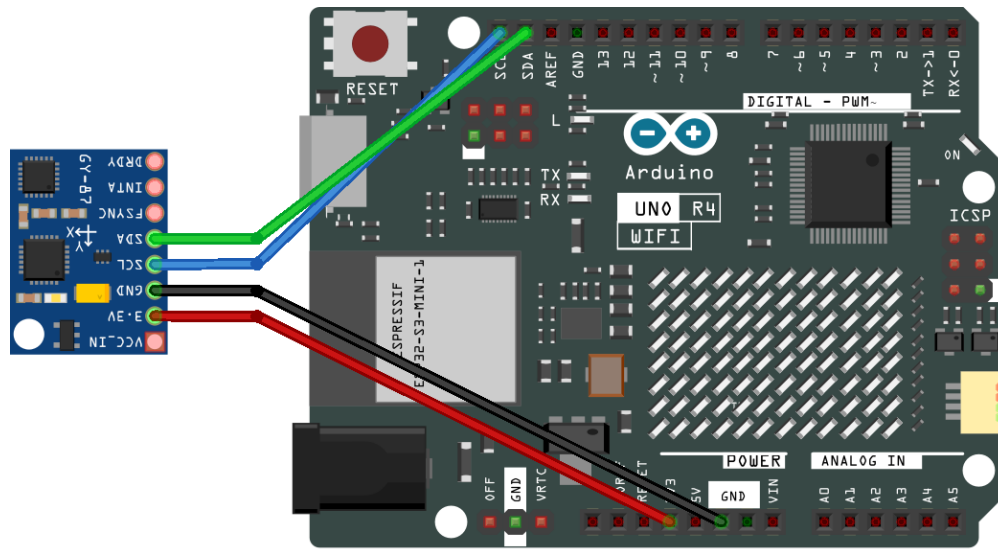
8.2 Wie scannt und erkennt man I2C-Adressen?

Dieses Tutorial nimmt das Scannen der I2C-Adresse des GY-87-Moduls als Beispiel und leitet Sie an, wie Sie I2C-Adressen erkennen können.

8.2.1 Verdrahtung

Verbinden Sie den SCL des GY-87-Moduls mit dem SCL des UNO R4 und verbinden Sie den SDA des GY-87-Moduls mit dem SDA des UNO R4.

Eine andere Möglichkeit besteht darin, den SCL des GY-87-Moduls mit A5 des UNO R4 zu verbinden und den SDA des GY-87-Moduls mit A4 des UNO R4 zu verbinden.



8.2.2 Code hochladen

Kopieren Sie den untenstehenden Code in Ihre Arduino IDE und laden Sie dann den Code hoch.

```
#include <Wire.h>

// Set I2C bus to use: Wire, Wire1, etc.
#define WIRE Wire

void setup() {
  WIRE.begin();

  Serial.begin(9600);
  while (!Serial)
    delay(10);
  Serial.println("\nI2C Scanner");

  // Enable bypass Mode for mpu6050
  Wire.beginTransmission(0x68);
  Wire.write(0x37);
  Wire.write(0x02);
  Wire.endTransmission();

  Wire.beginTransmission(0x68);
  Wire.write(0x6A);
  Wire.write(0x00);
  Wire.endTransmission();

  // Disable Sleep Mode
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0x00);
  Wire.endTransmission();
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

void loop() {
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for (address = 1; address < 127; address++) {
    // The i2c_scanner uses the return value of
    // the Write.endTransmission to see if
    // a device did acknowledge to the address.
    WIRE.beginTransmission(address);
    error = WIRE.endTransmission();

    if (error == 0) {
      Serial.print("I2C device found at address 0x");
      if (address < 16)
        Serial.print("0");
      Serial.print(address, HEX);
      Serial.println(" !");

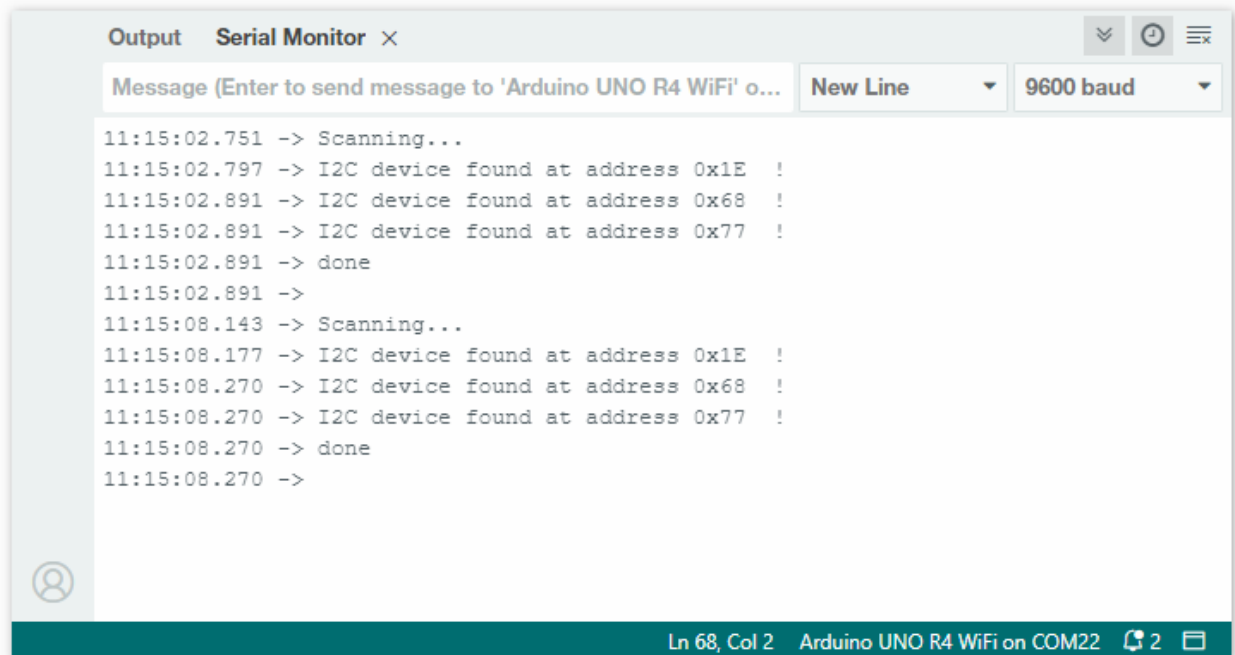
      nDevices++;
    } else if (error == 4) {
      Serial.print("Unknown error at address 0x");
      if (address < 16)
        Serial.print("0");
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000); // wait 5 seconds for next scan
}

```

Nachdem Sie den Code hochgeladen haben, öffnen Sie den seriellen Monitor und stellen Sie die Baudrate auf 9600 ein. Überprüfen Sie die Ausgabe im seriellen Monitor.

Dies sind die erkannten I2C-Adressen. Sie können sich auf relevante Informationen beziehen, um zu bestimmen, welche Chips diesen Adressen entsprechen. In diesem Fall steht 0x68 für MPU6050 und 0x77 für BMP180. Die Adresse 0x1E ist für QMC5883L, und gelegentlich (aufgrund unterschiedlicher Produktionschargen) kann die Adresse von QMC5883L auch 0x0D sein.



The screenshot shows the 'Serial Monitor' window in the Arduino IDE. The window title is 'Output Serial Monitor'. The input field at the top contains the text 'Message (Enter to send message to 'Arduino UNO R4 WiFi' o...'. The dropdown menu is set to 'New Line' and the baud rate is '9600 baud'. The main text area displays the following output:

```
11:15:02.751 -> Scanning...
11:15:02.797 -> I2C device found at address 0x1E !
11:15:02.891 -> I2C device found at address 0x68 !
11:15:02.891 -> I2C device found at address 0x77 !
11:15:02.891 -> done
11:15:02.891 ->
11:15:08.143 -> Scanning...
11:15:08.177 -> I2C device found at address 0x1E !
11:15:08.270 -> I2C device found at address 0x68 !
11:15:08.270 -> I2C device found at address 0x77 !
11:15:08.270 -> done
11:15:08.270 ->
```

At the bottom right of the window, the status bar indicates 'Ln 68, Col 2', 'Arduino UNO R4 WiFi on COM22', and '2' messages.

Vielen Dank!

Wir bedanken uns herzlich bei allen, die unsere Produkte erworben haben, bei den Evaluatoren, die sorgfältig bewertet haben, was wir anbieten, bei den Branchenveteranen, die uns wertvolle Anleitungen für unsere Tutorials gegeben haben, und bei unseren treuen Nutzern, die uns kontinuierlich folgen und unterstützen.

Ihr aufschlussreiches Feedback ist entscheidend, um uns zu motivieren, noch bessere und hochwertigere Produkte zu liefern.

Sollten Sie beim Einsatz dieses Kits auf Probleme stoßen oder Verbesserungsvorschläge haben, zögern Sie bitte nicht, uns unter service@sunfounder.com zu kontaktieren.

Für einen kollaborativen Ansatz empfehlen wir Ihnen, die Funktion [GitHub issue](#) zu nutzen. Diese Plattform ermöglicht es Ihnen, Dokumentationsprobleme direkt zu melden und sich mit unserer Gemeinschaft auszutauschen, um sicherzustellen, dass Ihre Stimme zur kontinuierlichen Verbesserung unserer Produkte beiträgt.

Urheberrechtshinweis

Alle Inhalte, einschließlich aber nicht beschränkt auf Texte, Bilder und Code in diesem Handbuch, sind Eigentum der SunFounder Company. Sie dürfen nur für persönliches Studium, Untersuchungen, Vergnügen oder andere nichtkommerzielle oder gemeinnützige Zwecke verwendet werden, entsprechend den geltenden Vorschriften und Urheberrechtsgesetzen, ohne die gesetzlichen Rechte des Autors und relevanter Rechteinhaber zu verletzen. Für jede Einzelperson oder Organisation, die diese zu kommerziellen Zwecken ohne Erlaubnis verwendet, behält sich das Unternehmen das Recht vor, rechtliche Schritte einzuleiten.